Leveraging Content from Open Corpus Sources for Technology Enhanced Learning

A Thesis submitted to the

University of Dublin, Trinity College

for the degree of

Doctor in Philosophy

Séamus Lawless

Knowledge and Data Engineering Group,

Department of Computer Science,

Trinity College,

Dublin

Submitted October 2009

Declaration

I, the undersigned, declare that this work has not been previously submitted as an exercise for a degree at this or any other University, and that, unless otherwise stated, it is entirely my own work.

_____

Séamus Lawless

October 2009

Permission to lend or copy

I, the undersigned, agree that the Trinity College Library may lend or copy this thesis upon request.

_____

Séamus Lawless

October 2009

ACKNOWLEGDEMENTS

ABSTRACT

As educators attempt to incorporate the use of educational technologies in course curricula, the lack of appropriate and accessible digital content resources acts as a barrier to adoption. Quality educational digital resources can prove expensive to develop and have traditionally been restricted to use in the environment in which they were authored. As a result, educators who wish to adopt these approaches are compelled to produce large quantities of high quality educational content. This can lead to excessive workloads being placed upon the educator, whose efforts are better exerted on the pedagogical aspects of eLearning design. The accessibility, portability, repurposing and reuse of digital resources thus became, and remain, major challenges. The key motivation of this research is to enable the utilisation of the vast amounts of accumulated knowledge and educational content accessible via the World Wide Web in Technology Enhanced Learning (TEL). This thesis proposes an innovative approach to the targeted sourcing of open corpus content from the WWW and the resource-level reuse of such content in pedagogically beneficial TEL offerings. The thesis describes the requirements, both educational and technical, for a tool-chain that enables the discovery, classification, harvesting and delivery of content from the WWW, and a novel TEL application which demonstrates the resource-level reuse of open corpus content in the execution of a pedagogically meaningful educational offering. Presented in this work are the theoretical foundations, design and implementation of two applications: the Open Corpus Content Service (OCCS); and the User-driven Content Retrieval, Exploration and Assembly Toolkit for eLearning (U-CREATe). To evaluate and validate this research, a detailed analysis of the different aspects of the research is presented, outlining and addressing the discovery, classification and harvesting of open corpus content from the WWW and open corpus content utilisation in TEL. This analysis provides confidence in the ability of the OCCS to generate collections of highly relevant open corpus content in defined subject areas. The analysis also provides confidence that the resource-level reuse of such content in educational offerings is possible, and that these educational offerings can be pedagogically beneficial to the learner. A novel approach to the sourcing of open corpus educational content for integration and reuse in TEL is the primary contribution to the State of the Art made by this thesis and the research described therein. This approach differs significantly from those used by current TEL systems in the creation of learning offerings and provides a service which is considerably different to that offered by general purpose web search engines.

TABLE OF CONTENTS

TABLE OF FIGURES

ABBREVIATIONS

ADL	Advanced Distributed Learning
AEHS	Adaptive Educational Hypermedia System
AH	Adaptive Hypermedia
ANN	Artificial Neural Network
CA	Content Aggregations
CGI	Common Gateway Interface
CML	Chemical Markup Language
CMU	Carnegie Mellon University
DEC	Digital Equipment Corporation
DESIRE	Development of a European Service for Information on Research and Education
DOM	Document Object Model
DRM	Digital Rights Management
EIT	Enterprise Information Technologies
GPL	General Public License
GUI	Graphical User Interface
HITS	Hyperlink Induced Topic Search
HTTP	Hypertext Transfer Protocol
idf	Inverse Document Frequency
IP	Intellectual Property
IPR	Intellectual Property Rights
IR	Information Retrieval
ISC	Internet Systems Consortium
ITS	Intelligent Tutoring Systems
LETSI	Learning, Education & Training Systems Interoperability
LO	Learning Object
LOM	Learning Object Metadata
LRM	Learning Resource Metadata
MAP	Mean Average Precision
MERLOT	Multimedia Educational Resource for Learning and Online Teaching
MRR	Mean Reciprocal Rank

MVC	Model View Controller
NDLR	National Digital Learning Repository
NIST	National Institute for Standards and Technology
OCCS	Open Corpus Content Service
ODP	Open Directory Project
OER	Open Educational Resource
RBSE	Repository-Based Software Engineering
RDF	Resource Description Framework
RDV	Representative Document Vector
RTFS	Rich Full-Text Search
SCC	Strongly Connected Component
SCO	Sharable Content Object
SCORM	Sharable Content Object Reference Model
SOA	Service Oriented Architecture
SOAP	Simple Object Access Protocol
SURT	Sort-friendly URI Reordering Transform
SUS	System Usability Scale
Swish-e	Simple Web Indexing System for Humans - Enhanced
TEL	Technology Enhanced Learning
tf	Term Frequency
TLU	Threshold Logic Unit
TRDR	Total Reciprocal Document Rank
TREC	Text Retrieval Conference
U-CREATe	User-driven Content Retrieval, Exploration and Assembly Toolkit for eLearning
UTF	Unicode Transformation Format
VSM	Vector Space Model
VUE	Visual Understanding Environment
WSDL	Web Services Description Language
WWW	World Wide Web
XML	Extensible Markup Language
ZPD	Zone of Proximal Development

1 Introduction

1.1 Motivation

The growing adoption of technology enhanced learning (TEL) in mainstream education is being driven by the potential to provide improvements in (*i*) the quality and flexibility of learning experiences; (*ii*) the ease of access to educational materials; [Wade et al. 04] (*iii*) equitable access to education across all areas of society; (*iv*) the cost-effectiveness of the educational process [Collins & Moonen 01]. Educational policy makers have also issued guidelines which promote an increase in the utilisation of educational technologies [DFES 03] [DFES 05] [NDP 01].

As learning evolves from static tutor-centric delivery mechanisms to interactive learner-centric experiences, educational technologies are seen as a method of facilitating dynamic interaction with the knowledge domain as a support to traditional teacher-driven methods of education [Laurillard 93] [Carey 93]. However, as educators attempt to incorporate the use of educational technologies in course curricula, the lack of appropriate and accessible digital content resources acts as a barrier to adoption [Brusilovsky & Henze 07]. Quality TEL resources can prove expensive to develop and have traditionally been restricted to use in the environment in which they were authored [Boyle 03]. As a result, educators who wish to adopt these approaches are compelled to produce large quantities of high quality educational content. This can lead to excessive workloads being placed upon the educator, whose efforts are better exerted on the pedagogical[1] aspects of TEL design [Cristea & Carro 07] [Dagger et al. 04] [Dagger et al. 05]. The accessibility, portability, repurposing and reuse[2] of TEL resources thus became, and remain, major challenges [Duval 01] [Koper 03] [Littlejohn 05].

[1] In this thesis the term pedagogy is used to refer to strategies or styles of instruction in education. The term is literally translated from the greek "to lead the child". Andragogy can be used when referring specifically to adult education, however in this thesis pedagogy will be used regardless of learner age.

[2] The *reuse* of digital content refers to the integration of existing content within a new application or educational offering. Resources can also be *repurposed* before being reused. This involves the manipulation, alteration, slicing or reformatting of existing content for use outside of the original context for which it was authored.

The key motivation of this research is to enable the utilisation of the vast amounts of accumulated knowledge and educational content accessible via the World Wide Web (WWW) in TEL. Through the identification and development of dynamic means of content discovery, classification and delivery, a scenario can be created whereby web-based educational content can be made available for incorporation into TEL experiences. This would enable educators and the developers of TEL offerings to concentrate on the pedagogical design of such offerings, rather than content authoring.

Picture a scenario where a lecturer in a university delivers an introductory course on human anatomy. The course is supplemented by a TEL exercise which examines the muscles of the body. The lecturer wishes to support his students as they progress, through the provision of accompanying digital content which provides information on all the muscles of the body in varying levels of detail. This content can be consulted by the student as they conduct the exercise. However, the lecturer does not want the students to use a general purpose search engine. As this is an introductory course, the students have not yet developed sufficient knowledge to confidently filter through content which potentially contains off-topic, inaccurate or poor quality content. Not to mention the risk of students being distracted by browsing the open web. This research aims to generate caches of subject-specific content for the educator through the implementation of information retrieval techniques and technologies. These caches of content can subsequently be made available via a search interface within the TEL application which delivers the exercise.

The increasing pervasiveness of the internet is fundamentally changing how people author, interact with and consume content. There are early signs of a shift in the way digital content is created, from the linear authoring and publication of material to the aggregation and reuse of existing content from various disparate sources. This trend, while still in its infancy, is apparent in the emergence of digital content repositories, learning objects, mash-ups and content aggregators. Many countries have begun to invest in national digital content repositories, in an attempt to encourage educators to share learning resources. The Multimedia Educational Resource for Learning and Online Teaching (Merlot) [Merlot] in the United States of America, Jorum [Jorum] in the UK and the National Digital Learning Repository (NDLR) [NDLR] in Ireland are just three examples of such repositories. Educational institutions are also beginning to open access to their learning resources.

OpenLearn [OpenLearn] and OpenCourseWare [OCW] are two such examples. However, these initiatives have encountered problems with user engagement, resulting from issues such as content repurposing and reuse, digital rights management (DRM) and institutional culture [Zastrocky et al. 06] [Ferguson et al. 07].

Learning objects (LO) are digital, self-contained 'chunks' of learning content [Wiley 00]. LOs aim to enable content reuse outside the context in which it was created and dynamic, 'on the fly' sequencing of resources [Farrell et al. 04]. However, this potential has yet to be fully realised [Weller et al. 03]. Specifications and standards which target content reuse are also emerging. These include the Shareable Content Object Reference Model (SCORM) [SCORM] and Learning Object Metadata (LOM) [LOM]. Creative Commons [Creative Commons] is a legal framework which provides free tools that enable content authors to share and distribute their work while protecting their copyright. Despite the emergence of these initiatives which address individual segments of the overall problem, content reuse remains an extremely challenging field of research.

TEL environments are attempting to respond to the demand for interactive, personalised learning experiences by providing increased support for functionality such as personalisation, adaptivity [Brusilovsky & Peylo 03] and dynamic learning object generation [Brady et al. 05] [Farrell et al. 04]. The incorporation of such flexibility and individual support is being heralded as one of the grand challenges of next generation TEL systems as it enables greater effectiveness, efficiency and student empowerment [Brusilovsky 04a]. As TEL systems attempt to support such features, one of the most significant problems encountered is their traditional closed corpus nature; there is an inherent reliance upon bespoke, proprietary educational content [Aroyo et al. 03]. This is particularly the case in Intelligent Tutoring Systems (ITS) where personalisation controls are often embedded in the content [De Bra et al. 99] [Conlan et al. 02a]. However, in more recently developed systems, which have become collectively known as the second generation of adaptive TEL systems [Brusilovsky 04a], there has been a significant shift towards the separation of personalisation and adaptivity information from the physical learning content. This development provides the opportunity for TEL systems to incorporate content from numerous diverse sources into educational offerings, yet the majority of these systems still source content from proprietary

repositories of learning resources. Examples of such systems include Knowledge Tree [Brusilovsky 04b], Aha! [De Bra et al. 03] and APeLS [Conlan & Wade 04].

To scalably support such next generation functionality as personalisation and on-demand adaptivity in mainstream education, TEL systems require access to large volumes of educational content which is varied in structure, language, presentation style, etc. [Brusilovsky & Henze 07]. Correspondingly, if the integration of educational technologies in course curricula is to become widespread, educators will need to be persuaded that the benefits provided by these systems outweigh the perceived negative effects, such as the manual effort required on content authoring and the unsettling effect of cultural change. If TEL environments can provide educators with functionality such as personalisation, combined with access to large quantities of quality educational content, the balance could begin to favour more widespread adoption of TEL. However, these TEL environments must subsequently provide methods of repurposing and reusing this educational content.

The WWW provides access to open corpus educational content in wide varieties and on a vast scale. Open corpus content can be defined as any content that is freely available for non-commercial use by the general public or educational institutions. Such content can be sourced from web pages, scholarly research papers, digital content repositories, commercial training repositories, forums, blogs, etc. However, this vast, accessible resource has yet to be thoroughly exploited in the field of TEL [Brusilovsky & Henze 07] and it has become increasingly difficult for educators to find relevant and useful educational content on the WWW [Ring & MacLeod 01]. If this open corpus knowledge and information is to be leveraged for use in TEL systems, methods of surmounting the heterogeneity of web content must be developed.

The wealth of possibilities for learner engagement provided by next generation TEL functionality can lead to the unintentional neglect of pedagogical requirements [Laurillard 07]. The metrics used to evaluate educational technologies have traditionally been technical in nature rather than focusing on the effectiveness of the learning conducted in the educational offering. It is of vital importance that such TEL systems and user interfaces address educational concerns and provide pedagogically meaningful learning experiences. TEL, as its name suggests, should enhance methods of learning through the use of

technologies. The design of these technologies should reflect pedagogically sound approaches to education.

This research aims to facilitate the targeted sourcing of open corpus content from the WWW and to support the resource-level reuse of such content in TEL offerings. There are several potential research areas which intersect with this research, but are deemed outside the scope of this thesis. It is not within the scope of this research to deliver adaptivity, personalisation or content repurposing within next generation TEL systems. The use of open corpus content in such systems has an impact upon their design, as the system has very limited apriori knowledge of the nature of each resource [Aroyo et al. 03]. The conversion of content to applicable granularities and formats for reuse within such systems will also need to be addressed by future research. It is also not within the scope of this research to investigate or implement means of resolving barriers to content reuse caused by Digital Rights Management or Intellectual Property Rights.

1.2 Research Question

The research question posed in this thesis is *what are the appropriate techniques and technologies required (i) to support the discovery, classification and harvesting of educational content from open corpus sources and (ii) to support educators and learners in the exploration, incorporation and reuse of such content in pedagogically meaningful TEL experiences*.

This research is focused on the design and development of a tool to support the classification and retrieval of high quality educational content from the WWW. The aim of this content service is to enable the discovery, classification, harvesting and delivery of educationally significant content from the WWW and digital content repositories and to make it available for use in TEL systems. The second focus of this research is the design and development of a user-driven TEL interface to demonstrate the integration of such content, harvested from open corpus sources, into a TEL experience. This interface should support both learners and educators in the exploration and utilisation of this content as a part of a pedagogically meaningful educational experience.

Many traditional TEL systems are restricted to the use of educational content which has been developed and structured in a proprietary fashion. The scalability of these systems is limited as a result of this dependence and the incorporation of new content becomes more difficult. Despite the emergence of standards which promote content reuse, such as SCORM, problems with lack of engagement and the complexity of implementation has limited widespread adoption. By removing this dependence upon bespoke content and supporting the incorporation of educational content from open corpus sources, the barrier to adoption of such TEL systems is removed. Educators, due to decreased content authoring workloads, can invest more time and effort on the pedagogical aspects of TEL offering design. This research is exploring the possibility of developing a framework for TEL which supports the discovery, classification and harvesting of educational content from open corpus sources and its incorporation into pedagogically meaningful TEL offerings.

1.3 Research Goals and Objectives

In answering the proposed research question, the objectives of this thesis are (i) to investigate, prototype and evaluate an approach for supporting the discovery, classification, harvesting and delivery of educational content from open corpus sources and (ii) to design, prototype and evaluate a TEL application which enables the incorporation and resource-level reuse of such content during the execution of a pedagogically meaningful educational experience. These two main objectives can be further disaggregated into the following specific set of research objectives:

1. The Discovery, Harvesting, Classification and Delivery of Open Corpus Content
    a. To investigate and specify techniques and technologies which can be used to enable the dynamic discovery, classification and harvesting of content residing on the WWW and in certain defined digital content repositories.
    b. To design and develop an application, based on these techniques and technologies, which enables the creation of caches of educational content, sourced from the WWW, defined by subject area.
    c. To investigate and implement the technologies required to index such open corpus content so that it is searchable and accessible to learners and educators.

2. TEL Application to Demonstrate the Resource-Level Reuse of Open Corpus Content

a. To identify the learning theories and technologies required to support pedagogically beneficial TEL experiences in an interactive, learner-driven TEL application.
b. To design and develop a demonstrator educational application which enables the learner-driven exploration and resource level reuse of content provided by the open corpus content service.

To achieve these research objectives there are certain core goals that must first be completed. An initial goal is to identify the current trends in the creation and management of digital content on the WWW, and specifically, the utilisation of such content by TEL systems. The growth of the WWW, and the emergence of tools which make content creation and distribution accessible to all users, has affected the traditional means of educational content authoring and publication. It is essential to review these changes and trends and examine how they may affect content utilisation in TEL.

Another goal of this research is to examine the use of pedagogical methods and educational theories in TEL. There is increasing, and understandable, demand to ensure the evaluation of TEL applications primarily measure educational performance. TEL applications need to be designed to deliver educational offerings grounded in traditional pedagogical frameworks. In order to ensure that the TEL design conducted by this research is appropriate and beneficial, a review of educational theories and their relationships to educational technologies must be completed.

A further goal of this research is to identify current methods of information retrieval on the WWW. As the WWW continues to expand it will become increasingly difficult to locate and aggregate large amounts of quality content in a specific subject area. Techniques and technologies from the Information Retrieval community, such as web crawling and content indexing, can be applied to address this problem. To examine these techniques and technologies in more detail, a state of the art review and appraisal of Information Retrieval on the WWW will be conducted.

This research extends existing and proven approaches to content retrieval to establish an educational content service that can discover, classify, harvest and deliver content sourced

from the WWW to learners and educators within TEL scenarios. Based upon the development of this content service an integral goal of this research is to create a supporting educational interface to illustrate, demonstrate and validate that it is possible to reuse open corpus content in a pedagogically meaningful way. This interface should allow learners to explore open corpus content and incorporate it into a TEL experience. This interface will implement the concept-level reuse of resources; it is not a content repurposing or slicing tool. When designing and developing these applications it would be neither practical nor logical to re-invent tools which address individual aspects of the desired functionality. As such, it is a goal of this research, where possible, to incorporate and combine existing open source solutions.

In order to successfully address the goals and objectives set forth in this thesis, an in-depth evaluation of both the process of content discovery and the support given to learners and educators through the user interface is performed.

1.4 Research Contribution

This work makes two notable contributions to the state of the art of technology enhanced learning. These contributions are illustrated throughout this thesis. Firstly, this research proposes a novel approach to the sourcing and utilisation of educational content in TEL. While web-based content is used in individual educational scenarios in TEL, the scale of open corpus content available remains largely unexploited. It is not common practice to create large collections of related content which can be utilised by TEL applications in the scalable delivery of learning offerings in mainstream education. In this research, content residing in open corpus sources, such as the WWW and digital content repositories, is made available to TEL systems through a novel tool chain which seamlessly integrates methods of discovery, language identification, classification, harvesting and storage. Subject-specific content caches are indexed and made searchable via a web interface to enable exploration and utilisation by the learner or educator [Lawless et al. 2008a].

A second contribution to the state of the art is a user-driven TEL interface, which was prototyped and developed to provide an environment which demonstrates and validates the utilisation of open corpus content in pedagogically meaningful learning experiences. This interface exploits the novel content retrieval tool chain to enable the exploration and

resource-level reuse of open corpus educational content. This TEL interface was developed to reflect elements of numerous theories of learning, including Cognitivism, Constructivism, Behaviourism and Enquiry-based Learning. The current generation of TEL systems are closed corpus in nature, relying upon bespoke, proprietary educational content; this interface illustrates that it is possible to leverage open corpus educational content, sourced from the WWW and digital content repositories, in TEL application design [Lawless et al. 2008b].

1.5 Research Approach

To accomplish the research goals and objectives identified above, it was necessary to provide an overview of pedagogical approaches in education with a specific emphasis on the design of TEL applications. This detail is located in chapter two of this thesis. This chapter begins with an introduction and examination of the main theoretical categorisations of the process of learning. Methods of mapping these pedagogical approaches into TEL design are then outlined. Following this is a review of the current trends of content authoring and publication on the WWW and the methods of content utilisation in TEL. This is achieved by firstly, analysing the various emerging techniques and technologies in these areas. Based upon this analysis, the success of these developments and their potential applicability in education is discussed.

A review of Information Retrieval on the WWW is conducted, specifically focusing on the areas of web crawling and web-based search. This review can be found in chapter three of this thesis. It provides the reader with a detailed insight into the foundations and current trends in these areas. A detailed description of web crawling and web-based IR algorithms is provided, followed by an examination of the growth of the WWW. These sections act as an introduction to, and foundation for, an analysis of focused crawling techniques and technologies. This is followed by a review of content indexing and retrieval in relation to web-based content. Based on this analysis, a discussion is provided on the elements of web-based IR which could be used to influence the discovery and collation of subject-specific educational content.

The design and architectural vision of this research is then illustrated in chapter four of this thesis. Base requirements for both educational content retrieval and an illustrative open-corpus educational environment are detailed. Based on influences from chapter two, design

requirements were developed which are guided by the educational theories discussed. This provides the educational foundation for the design of a learner-driven TEL interface. Based on influences from chapter three, technical requirements were developed for the discovery, classification, harvesting and delivery of educational content sourced from the WWW.

The technical implementation of a novel application tool chain, called the Open Corpus Content Service (OCCS), is then provided in chapter five. The OCCS is responsible for the discovery, classification, harvesting, storage and indexing of educational content from open corpus sources. This includes a description of the architecture of the OCCS and the different techniques and technologies which it uses. Based on the design specifications in chapter four, the components of the OCCS, and their integration, are described. The implementation of a learner-driven TEL interface, called the User-driven Content Retrieval, Exploration and Assembly Toolkit for eLearning (U-CREATe), is then detailed. This interface demonstrates the validity of open corpus content utilisation in TEL. Based on the usability requirements discussed in chapter four, the features of U-CREATe are described.

This thesis then proceeds to present the results of the trial and evaluation of this research. To align with the research question, stated in section 1.2, the OCCS was trialled with subject domain experts. This trial focussed on, and analysed, OCCS performance and usability and also content quality and accuracy. This evaluation is located in chapter six. The chapter also details trials of the U-CREATe system in combination with the OCCS with current students of Trinity College Dublin. The analysis of these trials focuses on several aspects of the research including interface usability, student performance and knowledge gain.

This thesis concludes with a description of the objectives and achievements of this research. A summary of key contributions to the state of the art of technology enhanced learning which are attributed to this research are provided. Finally, a discussion of the pertinent future work to continue and grow this research is presented.

2 Technology Enhanced Learning and Educational Content

2.1 Introduction to Technology Enhanced Learning

The process of learning in formal education no longer takes place solely in traditional, educator-centric settings. Interactive learner-centric experiences are being used to support learner collaboration, knowledge acquisition and reflection. Learner enquiry, activity and engagement are key requirements in such experiences and TEL applications are being designed and utilised to meet these requirements [Wade & Ashman 07]. In addition to delivering such learner-centric educational experiences, TEL can also be used to supplement and support more traditional educator-driven approaches to education. As TEL applications attempt to exploit the opportunities provided by new technologies, the requirements of the pedagogical frameworks implemented and the aspirations of the designer of the educational experience must be satisfied [Laurillard 07].

Traditionally, TEL applications have generated learning offerings using caches of educational resources, developed solely for that specific context and application. The system is aware of the format and content of each individual resource in advance, and has explicit knowledge of any relationships between resources [Aroyo et al. 03]. These educational resources, which conform to proprietary structures, are not easily portable or reusable outside of the specific TEL application or even in other educational contexts. The mainstream adoption of the WWW and new initiatives which make content creation and distribution open and easy, have altered the entire landscape of educational content authoring and publication. TEL systems are beginning to evolve in an attempt to reap the benefits of these changes, creating the potential for content reuse across educational contexts, application domains, institutions and even geographical boundaries.

This chapter examines these trends and issues with respect to educational content and technology enhanced learning. A categorisation of learning theory and related pedagogical approaches are discussed and methods of mapping these approaches into the design of pedagogically effective TEL applications are explained. Recent changes in the creation, management and distribution of digital content and how these changes can be applied in TEL

are then analysed. The current methods of educational content utilisation in TEL systems are also examined.

2.2 Educational Theories in Technology Enhanced Learning

Traditionally, it was common practice for the evaluation of educational technologies to be based upon the technical performance of specific instructional delivery methods rather than the pedagogical effectiveness of the educational offering. However, there has been growing demand in the research community for the evaluation metrics of technology enhanced learning to be primarily educational, rather than technological [Pittard 04] [Jochems et al. 04]. As a result, extensive research has been conducted into mapping pedagogical theory to educational practice in an effort to deliver better pedagogically-informed TEL design [Mayes & de Freitas 04]. It should be understood that the role of technology in such educational environments is to deliver enhancements to the educational process, which is nonetheless grounded in effective pedagogical frameworks.

Good pedagogical design should ensure that there are no inconsistencies between the curriculum being taught, the teaching methods employed, the learning environment used and the assessment procedures adopted [Biggs 07]. This means that the intended learning outcomes of the educational offering influence the learning and teaching activities that are used. They also influence the assessment measures which test if the outcomes have actually been achieved. However, to ensure that the learning and teaching activities selected can achieve the learning outcomes required, there must be some mapping between learning theory, pedagogical approach and the technology enhanced learning implementation.

The educational validity of TEL systems must be ensured, now more than ever, as a new generation of learners begin to participate in the educational experiences they offer. This generation of learners, sometimes termed “Digital Natives” [Prensky 05], have grown up interacting with digital technology and are much more comfortable interacting online and performing multiple activities simultaneously. These learners will not settle for inflexible systems which restrict how they conduct their learning or the types of content they can use to learn. The pedagogical design and assessment measures employed in such systems thus become extremely important.

It is the aim of this section to broadly introduce the most important areas of educational theory which can influence the design of TEL applications. A theoretical categorisation of learning is introduced and examined. Methods of mapping the pedagogical approaches contained in these categories, into TEL design are then outlined. This section is intended as a general introduction to what is a very complex and very active area of research. Some of the approaches discussed below, influenced the design of elements of this research detailed in chapter 4.

2.2.1 Theoretical Categorisations of Learning

One approach to the categorisation of learning at a theoretical level is to divide the process of learning into three broad and overlapping perspectives [Greeno et al. 96] [Mayes & de Freitas 04]. These perspectives are:

- The associationist/empiricist perspective, which defines learning as an activity.
- The cognitive perspective, which defines learning as achieving understanding.
- The situative perspective, which defines learning as a social practice.

2.2.1.1 Associationist/Empiricist Perspective

In this approach, knowledge is defined as a collection of skill-components and their associations. Learning is the process of connecting these units through sequences of activity [Putnam 95]. The learning theories of behaviourism, associationism and empiricism are contained within this category.

A behaviourist learner is perceived as a passive recipient of knowledge [Skinner 75] [Skinner 77]. Learning is then defined as the acquisition of this knowledge through activity combined with positive and negative reinforcement [Tuckey 92] [Carlile et al. 04]. Behaviourism has been widely dismissed as a serious theoretical basis for education. This is partly due to the mistaken belief that it can only be implemented in educator-centric educational approaches. In fact behaviourism promotes active learning-by-doing coupled with an emphasis on feedback from the educator to ensure the reinforcement of educationally beneficial behaviour [Wilson & Myers 00]. While behaviourism allows little room for creativity or independent learning, it does promote beneficial practices such as repetition in learning and the use of strong and varied stimuli to avoid lack of engagement [Carlile & Jordan 05].

Empiricism [Kolb 84] builds upon the notion of active learning and stresses the role of experience in the process. This approach defines learning as a cycle combining concrete experiences and reflections upon these experiences. In associationist approaches [Gagné 85] learning activities and knowledge units are arranged in sequence based upon their relative complexity. The simpler knowledge components are then assigned as pre-requisites to the more complex components. This creates sequences of instruction [Mayes & de Freitas 04] whereby the learner can progress by learning in small, logically ordered steps. It should be noted that the validity of the assumption that knowledge needs to be taught in specific conceptual sequences has been questioned [Resnick & Resnick 92]. The associationist approach, while still widely used across all domains of education, is more ideally suited to childhood education or to the teaching and assessment of competencies. It is not a model which is well suited to andragogical methods, particularly university education and higher level learning.

2.2.1.2 Cognitive Perspective

The cognitive approach attempts to model the mental process of interpreting information and constructing meaning [Newell 90]. Knowledge is viewed as a collection of conceptual models, constructed in the mind of each individual. Knowledge acquisition is the result of interactions between an individual's experiences and the conceptual models which have already been created. Learning is defined as the development of "understanding" through the construction and refinement of such mental models. This is in sharp contrast to the associationist view of passive information processing and association forming which disregards mental processes.

Bloom defined learning using three categories; the cognitive, the affective and the psychomotor [Bloom & Krathwohl 56]. In his model, the cognitive domain describes understanding or "knowing". The affective domain relates to attitudes and emotions and the psychomotor domain relates to actions and the ability to physically manipulate objects. When describing the cognitive domain, Bloom classified thinking using a taxonomy of six distinct cognitive layers of complexity. Bloom's Taxonomy has become a standard for classifying learning outcomes and has recently been revised and updated [Anderson et al. 01]. For more detail on the cognitive aspect of Bloom's Taxonomy please see Appendix A.

The process of learning as defined by the cognitive approach is consistent with constructivist pedagogical methods. Constructivism [Piaget & Inhelder 69] is an approach to pedagogy in which learners develop "constructs of understanding" by building upon their existing knowledge and previous experiences. New information is processed with reference to existing mental models and new, or adapted, knowledge constructs result. Learners are described as being particularly effective at constructing new knowledge when they are engaged in personally meaningful activities. In constructivist approaches, concepts can be considered as tools; something to be understood through use, rather than self-contained entities to be delivered through instruction [Brown et al. 89]. For more detail on constructivism please see Appendix A.

A similar approach to the construction of knowledge by the learner is that of Enquiry-Based Learning [Kahn & O'Rourke 05]. In Enquiry-Based Learning, the learner is engaged with a problem or scenario that is sufficiently open-ended to allow a variety of responses or solutions. The learner decides upon and steers the methods of enquiry. This requires the learner to utilise existing knowledge models and identify their learning needs. This approach stimulates curiosity in the learner and encourages them to actively seek solutions to the problem. Responsibility for analysing information and presenting possible solutions falls on the learner.

Cognitivists attempt to model the means by which learners acquire and organise their knowledge. Pedagogical approaches are then designed to exploit these patterns. For instance the relationship between information processing and memory is often exploited to aid knowledge retention [Carlile et al. 04]. Information received by the learner in the form of sensory input or stimuli must be passed through short term memory and "encoded" before it can be stored in long term memory. This tends to produce two categories of learners; "surface learners" who attempt to retain information in short term memory, usually due to information-overload, and "deep learners" who by attempting to understand the concepts, can more easily encode the information and transfer it to long term memory. Surface Learners are generally unsuccessful in retaining information and thus fail to acquire new knowledge effectively. Mind mapping [Buzan 74] is an approach which can be used to aid learner retention by utilising multiple media to present information. Mind mapping can also be used

in constructivist educational systems to help the learner reflect upon and refine their knowledge models. For more detail on mind mapping please see Appendix A.

2.2.1.3 Situative Perspective

In the situative perspective, an underlying assumption is that the learner will always be affected by the social and cultural setting in which the learning takes place, and that this situation will at least partly define the learning outcomes of an educational experience. This pedagogical approach focuses on the social distribution of knowledge [Mayes & de Freitas 04]. Activity, motivation and learning, which form the basis of the cognitivist approach to education, are all related to the learner's need for a positive sense of identity and self-esteem, both of which are shaped by social forces.

There are two primary facets of situated learning [Barab & Duffy 99]. The activity-based view emphasises context-dependent learning in informal settings. In this approach, learning activities should be authentic to the social context in which the knowledge is normally embedded. The most important design feature of such tasks is the relationship between the nature of the learning activity in an educational environment and how the skills acquired will be used in real life. The second view places more emphasis on the learner's relationships with the people in specific communities of practice. Described as an "environment of apprenticeship" such communities enable learners, who are initially involved only peripherally, to become more actively involved in the community of practice through observing and participating in activities [Lave & Wenger 91]. Learners progress from initial low-risk involvement to more legitimate contribution to, and participation in, community practices.

As with all situative approaches to learning, a key underlying element of the theory of social constructivism is that learning, and the development of knowledge, is directly influenced by the environment in which the learning occurs and the people with whom the learner is interacting. As described in Appendix A, social interaction and communication can be a key facilitator of learner reflection [von Glasersfeld 95], which is an important element of constructivism. Similarly, activity theory focuses not on the individual learner, but on a group, or social unit, of learners called an activity system. An activity system consists of a group, of any size, pursuing a specific goal in a purposeful way [Peal & Wilson 01].

The research of the Russian Psychologist Lev Vygotsky [Vygotsky 34] laid the foundations for situative approaches to learning such as social constructivism and activity theory. This work stressed the importance of individuals acting as mediators in the learning process. Vygotsky defined the "Zone of Proximal Development" (ZPD) which describes the relationships between learning, human interaction and learner development. The ZPD is defined as the distance between a learner's current conceptual development, as measured by independent problem solving, and that learner's potential capability when receiving guidance, or "in collaboration with more capable peers" [Vygotsky 78]. In this theory, an individual can learn a certain amount in isolation, but when working with a expert guide or in collaboration with peers, a higher level of conceptual development can be achieved.

In such approaches, the educator provides guidance, often referred to as "scaffolding" [Carlile & Jordan 05] [Mayes & de Freitas 04], to support the learner in their attempts to improve their understanding of a particular concept. As the learner increases their knowledge over a period of time and becomes more competent, this support should be gradually withdrawn, allowing the learner to become more independent. The skills, knowledge, and cognitive tools developed and refined by this learning process can be internalised and used by the learner in future problem-solving and self-directed learning [Peal & Wilson 01].

2.2.2 Mapping Educational Theory to the Pedagogical Design of Learning Environments

To ensure that the activities offered by learning environments can achieve the desired learning outcomes, there must be some mapping between the learning theories described above, and the pedagogical design of learning environments. Each environment must provide specific functionality which reflects the approach to knowledge acquisition of the learning theory, or theories, which are being used.

The associationist perspective emphasises highly focused sets of goals with clear and timely feedback from the educator. Associative learning environments should offer organised activities. These activities should ideally be sequenced as personalised pathways which reflect the individual's prior knowledge and performance. In associative environments educational material can be broken up into small, self-contained conceptual units. These units

can then be sequenced according to the desired learning outcome. Frequent feedback should be supported to ensure that beneficial behaviour is reinforced.

The cognitive perspective emphasises the development of interactive educational environments which promote the construction of understanding. Activities offered by such environments should encourage learner-driven exploration and experimentation aimed at the learning of broad principles. Cognitive environments should ensure that short term memory isn't overloaded by presenting too much information simultaneously. By presenting material in more than one form, these environments can help improve the transfer of knowledge to long term memory. Learners should be supported in reflecting upon topics already learned, this can improve knowledge retention.

The situative perspective emphasises the development of environments which facilitate learner participation in educational communities and encourage collaboration. These systems should provide activities which require the formulation and solving of realistic problems. The learner should be supported in developing an identity within a community of peers as a capable and confident participant. The fostering of educational relationships between individual learners should also be facilitated. A mapping between the ZPD approach to learning and the design of web-based learning environments has been proposed [Peal & Wilson 01] where the following features are employed:

- Learning activities that are part of real or simulated activity systems, with close attention to the tools and interactions characteristic of actual situations;
- Structured interaction among participants;
- Guidance by an expert;
- Eventual surrender of learning control to increasingly competent learners.

As Mayes and de Freitas propose [Mayes & de Freitas 04], the "modal pedagogical model" would combine these learning theories and define how to engage learners in meaningful activities, provide relevant and timely feedback, promote learner reflection, align assessment with learning outcomes and encourage the creation of communities of learners. A modal TEL model would define how technology would be used to implement each of these aspects within an educational environment.

Existing TEL models for learning environments typically exhibit characteristics of one or more of the pedagogical models described above. Few fit neatly into a single pedagogical perspective, but tend to overlap in some areas. Figure 2-1 below displays a sample mapping of some approaches to TEL onto the pedagogical strands discussed above. As there are overlapping aspects of the cognitive and situative perspectives, these have been disaggregated into three categories: cognitive/constructivist; socially-mediated constructivist; and communities of practice.

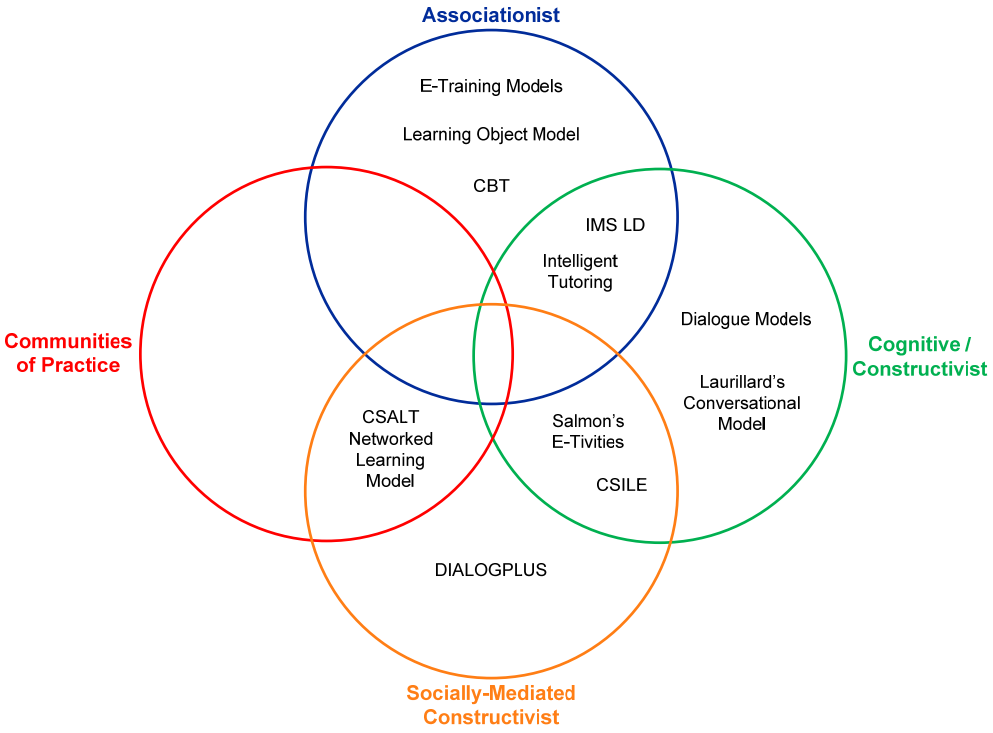


Figure 2-1 TEL Models by Pedagogical Category[3]

Four broad questions can be used in an attempt to categorise the pedagogical approach of an TEL model [Mayes & de Freitas 04]:

1. Are concepts divided into small self-contained units which are directly tied to learning outcomes?
2. Is ownership of the educational activity placed with the learner and does this activity produce outcomes upon which the educator can provide feedback?
3. Is active discussion promoted across communities of learners
4. Is there a focus on the development of real-world practice?

[3] This diagram is taken from [Mayes & de Freitas 04].

If the first question best describes an TEL model, then this maps most accurately onto the associationist category. If the second question is more characteristic of the TEL model used, then the cognitive/constructivist pedagogical approach is dominant. TEL models which are best described by question three map onto the socially-mediated constructivist category. Finally, TEL models most accurately described by question 4 are mapped onto the communities of practice category. However, it should be noted that these pedagogical strands are very high-level and numerous TEL models will display characteristics of multiple strands.

2.2.3 Analysis

As detailed in the above section, one method of categorising learning is to divide the various existing pedagogical approaches into three broad, overlapping categories: the associationist; the cognitive; and situative perspectives. The design and implementation of TEL environments should be influenced by one or more of these pedagogical categories. TEL applications quite often display characteristics of more than one learning perspective. They rarely fit neatly into a single approach to learning.

Objective 2-b (see section 1.3) of this research is to design and develop a demonstrator educational application which enables the learner-driven exploration and resource level reuse of content provided by an open corpus content service. During the design of such a TEL application, it should be ensured that the intended functionality of the application can be mapped back to approaches promoted by the established pedagogical categories discussed above. This ensures that the desired learning outcomes of experiences offered by the TEL application can be effectively achieved.

The use of TEL systems has become increasingly common in mainstream education and these systems have begun to provide more complex features and delivery methods. The wealth of functionality provided by these applications, and the resulting possibilities for learner engagement, can often lead to the neglect of pedagogical concerns within the educational experience. It is essential that the TEL application developed by this research primarily address educational concerns and deliver not only engaging, but pedagogically meaningful learning experiences. The evaluation of this application should also primarily focus on the educational outcomes rather than technological performance.

2.2.4 Summary

This section provided a broad introduction to the area of educational theory and pedagogy. A categorisation of a number of pedagogical approaches was described and illustrated. The theoretical and pedagogical approaches to education detailed in this section were discussed in relation to their influence on the design and implementation of TEL applications. Methods of mapping these pedagogical approaches to the design of TEL environments to ensure educational effectiveness were also outlined.

2.3 Educational Content Creation, Dissemination and Utilisation

Mainstream methods of digital content creation, management and distribution are rapidly changing and evolving. The emergence of the internet has heralded a revolution in the means by which information can be produced and shared. This is particularly true for the academic domain. The process of disseminating educational content was traditionally a linear process, beginning with the painstaking authoring of content by a domain expert and ending with the publication and sale of material in retail outlets or distribution through libraries. Information can now be authored by numerous individuals within physically distributed communities and dynamically aggregated to form collections of material in particular subject domains. Content can be instantly published to the WWW and, potentially, shared with millions of individuals worldwide at essentially no cost. This changing landscape of information authoring and management has enabled the emergence of initiatives such as digital content repositories, learning objects, mash-ups and content aggregators.

As the methods of digital content creation, management and distribution are evolving, so are the content utilisation approaches of educational environments. TEL systems are attempting to respond and adapt to the emerging forms of educational content authoring and dissemination by removing traditional dependencies upon bespoke, proprietary content. This creates the potential for cross-environment educational content repurposing and reuse.

The objective of this section is to provide the reader with an insight into the current trends in digital content creation, management and distribution and the current methods of content utilisation in TEL. It provides an analysis of various emerging techniques and technologies in these areas, their success to-date and their potential applicability in the TEL domain. Content modelling standards and Learning Objects are introduced. Educational digital content

repositories and related initiatives are then discussed. The section then continues with an introduction to emerging methods of content creation and dissemination on the WWW. The traditional, and current, methods of educational content utilisation in personalised TEL are then examined. The section concludes with a brief introduction to the issues surrounding Digital Rights Management and Intellectual Property Rights.

2.3.1 Learning Objects and Content Modelling Standards

There are many different definitions of what exactly constitutes a Learning Object (LO). LOs have been described as digital, self-contained 'chunks' of educational content [Wiley 00]. In the IEEE Learning Object Metadata (LOM) standard, LOs are defined as "any entity - digital or non-digital - that may be used for learning, education or training [Duval et al. 02]. Friesen describes LOs as "any digital resource with a demonstrated pedagogical value, which can be used, reused, or referenced to support learning" [Ring & MacLeod 01]. In another study [Bailey et al. 06], educators proposed the notion of a learning 'nugget' which represents a stand-alone learning activity which can vary in both size and scope. A learning nugget is primarily comprised of tasks which the learner will undertake in a defined context in order to achieve specific learning outcomes.

UNESCO [UNESCO] has adopted the term "Open Educational Resources" (OER) to describe the content infrastructure which enables the sharing of educational material, similar to the notion of LOs. The OER movement is attempting to deliver the open provision of educational resources, enabled by information and communications technologies, for consultation, use and adaption by a community of users for non-commercial purposes [Albright 05].

Regardless of precise definition, LOs attempt to facilitate the reuse of educational content, even outside the context or scenario in which they were created. These resources can theoretically include text, images, web-based resources, videos, animations, audio and any other medium which can be used to disseminate information. These resources can then be sequenced in a meaningful fashion to communicate or teach a concept. The notion of defined learning paths through collections of educational resources pre-dates the WWW [Davis et al. 93]. LOs also provide the potential for dynamic, 'on the fly' sequencing of educational resources [Farrell et al. 04]. However, such functionality has yet to be fully realised [Weller et al. 03].

The numerous, general definitions of LOs allow for large variations in the structure, granularity and purpose of an educational resource [Duval & Hodgins 03]. Content modelling standards, such as LOM, IMS Learning Resource Metadata (LRM) [IMS LRM], Dublin Core [Dublin Core] and the Sharable Content Object Reference Model (SCORM) [SCORM], attempt to address this problem. Such models provide a way of creating precise definitions of individual LOs and the means by which they can be reused and repurposed.

LOM [Duval et al. 02] is a conceptual data model which is used to define and structure metadata associated with a LO. This metadata describes the relevant characteristics of the educational content in the LO. The data model is divided into nine descriptive categories: general, life cycle, meta-metadata, technical, educational, rights, relation, annotation, and classification.

- The general category is used to describe the content of the LO. It provides information such as the title, language, description and coverage of the content. The general category also describes the functional granularity of the LO.
- The lifecycle category tracks the history and current state of the LO. Version detail is noted along with lists of contributors and their roles in the creation of the LO.
- The meta-metadata category describes the metadata instance itself, rather than the content of the LO.
- The technical category describes the technical requirements and characteristics of the LO such as format, size and installation requirements.
- The educational category describes the educational and pedagogic characteristics of the LO. This category is used to describe the modes of learning supported by the LO along with information on the intended user interaction, context and age-range of the content.
- The rights category describes the intellectual property rights and conditions of use of the LO. Details with regard to copyright restrictions and monetary fees are contained in this category.
- The relation category describes the relationship between the LO and other related LOs if any exist. There can be multiple instances of this category to define multiple relationships.
- The annotation category provides comments on the educational use of the LO.

- The classification category describes the LO in relation to a particular classification system such as the ACM or Library of Congress.

Learning object metadata descriptions are used in an attempt to improve the reusability of educational content, to aid its discoverability, and to facilitate the interoperability of individual LOs. Standards such as LOM typically rely upon the use of controlled vocabularies during the metadata authoring process. This ensures consistency in descriptions across multiple metadata instances and authors. This consistency improves the discoverability of resources when using information retrieval techniques.

The SCORM standard [SCORM 06] is known as a content aggregation model. It provides an XML binding for combining content modelling standards like LOM, and other content packaging standards such as IMS Content Packaging [IMS CP]. SCORM offers a means of representing individual pieces of educational content regardless of granularity and format, called assets, and methods of aggregating these assets to form a reusable object, independent of learning context, called a Sharable Content Object (SCO).

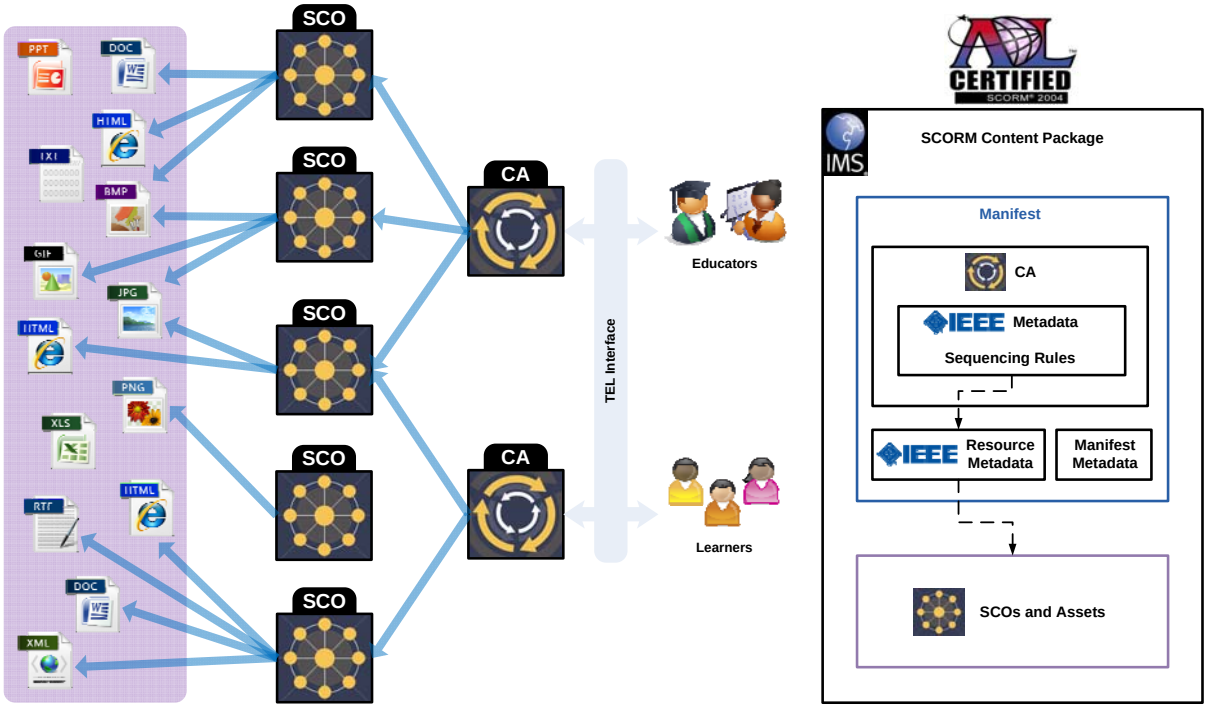


Figure 2-2 SCORM Content Aggregation Model

A SCO can be used within different TEL offerings to achieve different learning outcomes. Although a SCO can be formed from any collection of assets, they should ideally be of fine granularity, as this makes reuse more feasible [Verbert & Duval 04]. As illustrated in Figure 2-2 above, SCORM can also be used to define Content Aggregations (CA), which are maps used to organise SCOs into integrated educational resources such as modules or entire courses. SCORM uses LOM as the metadata standard which describes each asset, SCO and CA. An XML manifest file is used to describe all the elements of a SCORM LO and the content resources used by that LO.

2.3.1.1 Analysis

There is a divide in the TEL community with regard to the generation of metadata descriptions. Many feel that the sheer volume of content appearing on the WWW and in educational systems around the globe will necessitate the automated generation of metadata [Dill et al. 03] [Meire et al. 07]. However, there is concern and scepticism amongst some academics about the ability of automated metadata generation to produce quality, accurate descriptions of content [Dripps et al. 07] [NISO 04], and some believe metadata should only be manually created by subject domain experts. Despite these concerns, some recent evaluations have shown no statistical difference between the quality of metadata annotations which were automatically generated and those which were human authored [Meire et al. 07]. The majority of researchers in the field agree that the future of metadata generation most likely lies somewhere between these two extremes [Baird et al. 06].

While improving the reusability, discoverability and interoperability of LOs, the restricted vocabularies employed by metadata standards can act as barrier to user engagement with metadata authoring tools. They can be perceived as complex, restrictive or obscure by the end user [Bailey et al. 06]. This can reduce the level of adoption of the standards and the resulting volume of descriptions generated [Charlesworth et al. 07]. Users are often not willing to invest the time necessary to author detailed, descriptive metadata [Currier et al. 04]. A recent study has shown that in the opinion of educators, the ideal number of metadata fields to be populated is seven; name, description, tags, owner, permissions, type and language [Davis et al. 09]; The user also expected three of these tags to be automatically populated; type, owner and permissions.

There are limitations to content modelling standards such as LOM and SCORM. There is an inverse relationship between the potential reusability of a piece of educational content and its granularity. The larger and more complex a piece of content, the more specific the context it must be used in. Fine-grained pieces of educational content are much more easily reused outside of the context for which they were created. LOM was designed to describe course-grained LOs, even complete courses. This restricts the potential wide-scale reuse of LOM LOs. LOM is quite a complex metadata standard with a hierarchical structure of nine top-level categories and sixty individual metadata elements.

SCORM is also quite a large and complex standard. It relies upon LOM for the metadata description of individual assets, SCOs and CAs. It also requires content packaging and sequencing definitions. A SCORM package is defined using an XML manifest file, which groups and sequences the individual elements of the package and lists all the resources used. The manifest can even contain descriptions of the educational content. The fact that the manifest contains links to the actual content assets used, restricts the SCORM object to only using these specific resources. SCORM and LOM also rely upon the use of controlled vocabularies during the metadata authoring process. As discussed above, this can be perceived as restrictive and can further hinder user engagement.

The future development of SCORM has moved from ADL [ADL] to LETSI (Learning-Education-Training Systems Interoperability) [LETSI]. LETSI put out an open call for requirements for the development of SCORM 2.0 in July 2008 and held a workshop to define these requirements in October 2008. The new standard should be available in 2009.

The economic viability of LOs lies in the formalisation of a process which has always, historically, been an informal one; the sharing of educational resources among academics. Making digital LOs easily discoverable and accessible through initiatives such as digital content repositories can help to improve the process of content sharing and reuse. They advantages and disadvantages of such initiatives are discussed in the next section. If LO portability and reusability could be increased, it would help to balance the effort expended and rewards received by authoring LOs [Ring & MacLeod 01].

2.3.2 Digital Content Repositories

Although there were numerous motivational factors which contributed to the emergence of digital content repositories[4] in the education domain, it is possible to identify two main catalysts which resulted in the recent emergence of large volumes of institutional, regional and national repository initiatives.

The first main influencing factor in the growth of the digital content repository movement was the desire to spawn wide-scale content reuse. As a result of changing lifestyle factors, a growing number of people prefer the asynchronous flexibility of distance learning, or TEL courses offered by institutions through the WWW. Traditional classroom-based courses are also increasingly being supplemented with TEL modules [Wade et al. 04], due to demand and changes in educational policy. The growing popularity of such delivery methods has resulted in universities expending increasing amounts of time and resources in order to author high-quality courseware which is attractive to the learner [Boyle 03]. Digital content repositories were seen as a means to store and manage the content of such multimedia components effectively and efficiently.

The second prime factor was the quality of educational content entering the academic domain. As the publication of material on the WWW has become freely and openly accessible, the traditional means of preserving the quality of educational content through scholarly review has become less relevant, enforceable and achievable. The establishment of digital content repositories was an attempt to enhance information management and control, primarily in the realm of scholarly research and educational activities. Such peer-reviewed collections of educational resources were viewed as a means of ensuring the integrity and quality of educational content in the digital age.

The effort required to establish a digital content repository within an institution has been greatly reduced with the emergence of stable, open-source digital content repository solutions such as EPrints [EPrints], DSpace [DSpace] and Fedora [Fedora]. EPrints, initially developed in 2000 at the University of Southampton, is an OAI-compliant [OAI] repository solution

[4] Repositories can be used to collate and curate a wide range of materials and are not specifically related to academia. However, for the purposes of this thesis, the term repository is used to refer to a digital content repository which is used to store and share educational resources.

particularly focused on providing open access to an institution's peer-reviewed journal article output. DSpace is designed to provide the storage of, and access to, all of an institution's digital resources and supports a variety of data formats from text and images to video and data sets. Fedora claims to enable the storage, access and management of virtually any kind of digital content.

It has become increasingly difficult for educators to find relevant and useful educational content on the Web [Ring & MacLeod 01] so this filtration and control over the quality of resources being placed in repositories allows for more efficient access to high-quality information. Metadata has been used to supplement this process and describe the characteristics and content of such educational resources, as described previously in section 2.3.1. The application of metadata standards can help improve the precision of the search tools used in repositories. Digital repositories can be used to make educational resources available in a variety of media types to students, educators, and other researchers. This improves upon traditional scholarly journals and dissemination methods which relied upon purely textual means of communication.

Some early European research projects in the area of web-based TEL, such as ARIADNE [Forte et al. 96] and MTS [Graf & Schnaider 97] were based on content repository and reuse principles. MTS, developed as part of the EU-funded IDEALS project, investigated the modularisation of computer-aided educational offerings. Courses were divided into conceptual components and content could be selected for use in each component. ARIADNE consisted of multiple repositories of learning objects and associated metadata descriptions and an open set of tools to produce, index, and reuse this material.

Over the past decade increasing numbers of national digital content repositories have been developed. Such initiatives are viewed as a means to encourage educators from various educational institutions within a country, to collaborate upon the authoring of educational material and to share learning resources. The Multimedia Educational Resource for Learning and Online Teaching (Merlot) [Merlot] in the United States of America and the National Digital Learning Repository (NDLR) [NDLR] in Ireland are two such national repositories. In a study conducted in the UK in 2004 [JORUM 04], 88.6% of academic staff questioned were in favour of the establishment of a national repository of teaching materials, 79% said

they would contribute and 91% said they would use resources from the repository. In a more recent study, participants were asked "what type of repository would you be happiest to contribute to?" [Bates et al. 07]. A national subject-based repository was the most popular response at 49%. 17.9% responded with a general national repository and 16% said an institutional repository.

Educational institutions are also beginning to make their learning resources available through institutional repositories with open access. Examples of such initiatives are Carnegie Mellon University's Open Learning Initiative [CMU OLI], the Open University's OpenLearn [OpenLearn] and the University of Southampton's EdShare [EdShare]. OpenLearn, launched in October 2006, is designed to share the educational resources of the Open University. It is divided into the "LearningSpace" and the "LabSpace". LearningSpace provides formal, structured educational content from the Open University for use by learners regardless of level or need. Content is categorised by subject area and structured into learning units of between three and fifteen hours in length. LabSpace is a community-driven environment with promotes educational content sharing, reuse and repurposing. It incorporates all the content available in the LearningSpace and additional content which has been generated and submitted. OpenLearn claims to have over 5,400 learning hours of educational content up to postgraduate level in the LearningSpace and over 8,100 hours in the LabSpace. EdShare, launched in October 2008, allows educators and learners in the University of Southampton to collaborate on, or share educational resources. Resources can be made accessible to defined sets of individuals, to all members of a certain school or to the entire University. The author may also choose to make a resource openly accessible. As of March 2009 EdShare contained 622 resources consisting of 2477 individual items [Davis et al. 09], 33% of which had been made openly accessible.

The OpenCourseWare Consortium [OCW] is a collaborative group of over 200 higher education institutions and affiliate organisations from around the world. Each member of the consortium has its own OpenCourseWare implementation, which is a web-based publication of high-quality educational resources, organised as courses. The Massachusetts Institute of Technology (MIT) is a high-profile consortium member and has made the majority of its course content available via OpenCourseWare.

Professional societies such as the IEEE are also getting involved in the open access movement as a means of extending their global academic coverage. The IEEE signal processing society [IEEECNX] has recently joined forces with the Connexions project [Connexions] to develop a multi-lingual repository of peer-reviewed educational modules and courses related to signal processing, which are freely available. Connexions is an openly accessible repository platform that facilitates the sharing and reuse of small modules of content which can be pieced together to form courses.

OER Commons [OER Commons], launched in February 2007, adopts a meta-search engine approach to the discovery of freely available educational resources. Educational content is sourced from over 120 partners, including Carnegie Mellon University's Open Learning Initiative, Connexions and MIT's OpenCourseWare. A learner or educator can search for desired content and engage with the community of users through social bookmarking, tagging, rating, and reviewing.

JORUM [JORUM] is a JISC funded collaborative venture in UK Higher and Further Education to collect and share learning and teaching materials, allowing their reuse and repurposing. JORUM has approximately 300 registered institutions, of which 85 are active contributors of educational materials.

2.3.2.1 Analysis

Despite the numbers of digital content repository initiatives which have emerged over recent years, there is still a lack of engagement within the academic community. A mixture of cultural, legal and organisational issues must be addressed before digital content repositories can be accepted by the mainstream academic community as a means of content sharing, reuse and collaboration. Attitudes to openness, collaboration and sharing must change before engagement can be fully realised. Several hundred years of indoctrinated academic community dynamics cannot be discarded overnight and must evolve to incorporate these new approaches.

In September 2007, a JISC study [Charlesworth et al. 07] noted that there remained relatively little formal, large-scale sharing and reuse of content via digital repositories. In spite of the relatively rapid rate at which repository initiatives were being established, the volume of resources deposited in these repositories remains modest [Foster & Gibbons 05]. Repository

usage mainly occurred within funded projects which aimed to stimulate such activity [EFC 05]. Many projects are established with the intention of examining different approaches to knowledge dissemination. However, it has been found that often when such projects finish, this activity ceases [Guthrie et al. 08]. In many of the most well known national and international repository initiatives there remain insufficient volumes of contributed content to spawn regular and widespread content reuse.

For instance, the NDLR, a collaborative initiative between seven Irish Universities, fourteen Institutes of Technology and their affiliated colleges, established in 2004, currently contains only 1956 learning resources. In MERLOT, a national US repository, across all of areas of science and technology there are only 7837 educational resources. If this categorisation is narrowed down to Computer Science, the number is just 581. In Europe, the ARIADNE project was restricted by its reliance upon content authors submitting their content to the repository, and metadata experts manually indexing these learning objects once submitted. By October 2008, 12 years after the project initiation, the ARIADNE Knowledge Pool contained 4798 metadata descriptions of individual learning objects across all its subject domains. At the same point in time, Connexions, which was established in 1999, had 6934 reusable modules available for use across all subject domains[5].

Less than one third of the institutions registered on JORUM are active contributors of learning materials. JORUM state that the majority of content sharing to-date has occurred from within projects rather than institutions [Carter & Richardson 07]. JORUM also note that, in their experience, few higher education institutions are ready to instil widespread content sharing amongst their academics or have policies in place to actively encourage the sharing of educational resources. They state that "In terms of readiness for sharing TEL content, most organisations are not ready, with any plans not yet fully thought through and still in their infancy" [Charlesworth et al. 07].

The use of digital content repositories must be linked to a change in the way educational institutions function and are structured. The view in many institutions is that content repositories are merely a tool for content archiving and quality control, not to encourage

[5] Please note: The figures in this paragraph are accurate as of 18th of October 2008.

active content repurposing and reuse, collaboration and sharing of resources [Charlesworth et al. 07]. In many modern educational institutions, research output takes priority and there is a lack of encouragement and prestige associated with the sharing of teaching content. A recent JISC study [Margaryan 06] into learning object repositories and educational content sharing found that only 1.2% of respondents used repositories to share work in progress.

Educators can be motivated to engage with digital repositories by the promise of their expertise being promoted in other institutions and in the wider academic community. Attribution is a known motivator in educational research but as of yet has not been widely exploited in the teaching domain [Dripps et al. 06]. Cultivating a sense of prestige in relation to the quality and reusability of educational material is seen as a means of developing active educator participation in digital content repository initiatives. In a recent study [Bates et al. 07], 35% of respondents cited academic kudos as a reason for contributing to a repository. However, care must be taken to prevent a clash with the University desire for content to primarily promote the institution. In some educator surveys, correct author attribution is viewed as much more important than institution attribution, despite the institution often holding the copyright on the material [Bates et al. 06].

Despite this lack of mainstream engagement with digital repositories, there tends to be large volumes of small-scale, informal sharing, reuse and collaboration among peers. This is particularly evident in educators searching for content on the WWW and aggregating this content in numerous ways to create new resources which meet their own needs. [Davis et al. 09]. In many ways, the “build it and they will come” approach adopted by digital content repository initiatives, especially those born in academic institutions, has precluded systematic, in-depth studies of user behaviour and demand [Harley et al. 06].

The successful reuse of educational material partly relies upon the clear articulation of pedagogical intent. It is of the utmost importance that educators define the pedagogical purpose of resources placed in repositories. This will enable the reuse of the resource in suitable scenarios and also improve the discoverability of the resource in the repository [Charlesworth et al. 07]. This pedagogical definition could be achieved through the classification of the learning resource, using specific standards such as LOM or SCORM, by the educator when the resource is placed in the repository. However, as discussed previously

these standards tend to be quite complex and are more suited to learning resources of course granularity.

The learning resources in OpenLearn, and particularly in the LearningSpace, were typically written as individual components of a formal course, with integrated tuition, support and assessment using a specific style and medium of delivery [Lane 06]. Reusing this content outside of its original context or repurposing it for delivery in another context is a non-trivial task. Each learning unit available via the Learning Space is typically 3-15 hours of study time in length and has a single learning outcome. The extraction of a section of content from within a single unit could impair the pedagogical effectiveness of the educational resource.

As OpenCourseWare resources are structured as entire courses with very coarse granularity, the flexibility of content reuse is severely affected. Often educators repurpose specific sections of content or components of a resource and combine these with other content. In the case of OpenCourseWare offerings this is not a trivial task. Some research is being conducted into disassembling educational resources with a high level of aggregation such as OpenCourseWare courses [Pernías Peco et al. 08]. This is currently implemented in a semi-automatic fashion and shows promise as a means of improving the flexibility of content reuse.

Some academics are reluctant to contribute to digital content repositories as they are wary of the quality, style or format of their own content and are concerned about how it would be perceived and how useful it would be to a wider community of peers [Davis et al. 09] [Bates et al. 07] [White 06]. There can be a tendency to view content as not “flashy” enough or too simple. Authors are also concerned with the potential misuse of content [Charlesworth et al. 07].

It remains unclear if there are direct time and workload benefits to be gained from the use of digital content repositories. In a 2007 survey conducted on members of 98 UK Universities [Bates et al. 07] 44.3% of respondents disagreed or strongly disagreed that their workload was reduced due to the easy access to educational materials through a repository. 39.8% of respondents disagreed or strongly disagreed that it was clear how such materials could be used or reused. Author workload can be negatively impacted as educational content needs to

be frequently reviewed and updated to keep it accurate and relevant [Davis et al. 07]. This is particularly the case in areas such as science and engineering as techniques and technologies are developed and evolve at a rapid pace. In a study conducted using participants from the Open University and the University of Leicester, many tutors reported a lack of time for collaborative activities such as content sharing [Hewling 06]. While there was optimism about the possibility of shared resource repositories, it was felt that such initiatives must seamlessly integrate with the normal working life of academics.

Arguably the most significant barrier to engagement with digital content repository initiatives and educational content sharing is difficulties in relation to copyright law, ownership and intellectual property (IP). These issues will be briefly introduced in section 2.3.5 and warrant serious consideration and effort, however they are deemed outside the scope of this research.

2.3.3 Content Publication, Aggregation and Social Applications

The volume of content accessible via the WWW is currently experiencing explosive levels of growth, something which will be examined in more detail in section 3.4. This growth is largely due to emergence of a new generation of social platforms which have transformed user interaction and content production on the WWW. Blogs, wikis, photo sharing and social networking, to name but a few of these platforms, have opened up the WWW and allow any user to be a publisher of content. These platforms, combined with the distributed network structure of the WWW, have stimulated user-generated content creation, content sharing, collaborative content development, and the reuse and repurposing of content on the web. This change in emphasis and utilisation of the WWW has come to be known as Web 2.0 [O'Reilly 07]. The content available via many of these applications is potentially useful, sometimes unintentionally, in technology enhanced educational offerings.

The act of aggregating content can help make it easier to process. Instead of being confronted with a whole field of information, smaller, more logical subsets are grouped together [Porter 04]. As a result of the sheer volume of content on the WWW, content aggregation has become central to navigation. Following links from homepages through the navigation tree of a web site to reach the relevant content has become less commonplace, as illustrated in Figure 2-3 below. Instead, related content is collated on blogs, by search engines or in xml-based feeds which provide the user with direct access to the target content. As a result, pages of content are now more regularly constructed as self-contained units. It can no longer be

assumed that the content on a single page is automatically influenced by the context of the site as a whole. The user may have accessed the content directly and as a result, any context usually created by browsing the site is no longer applicable.

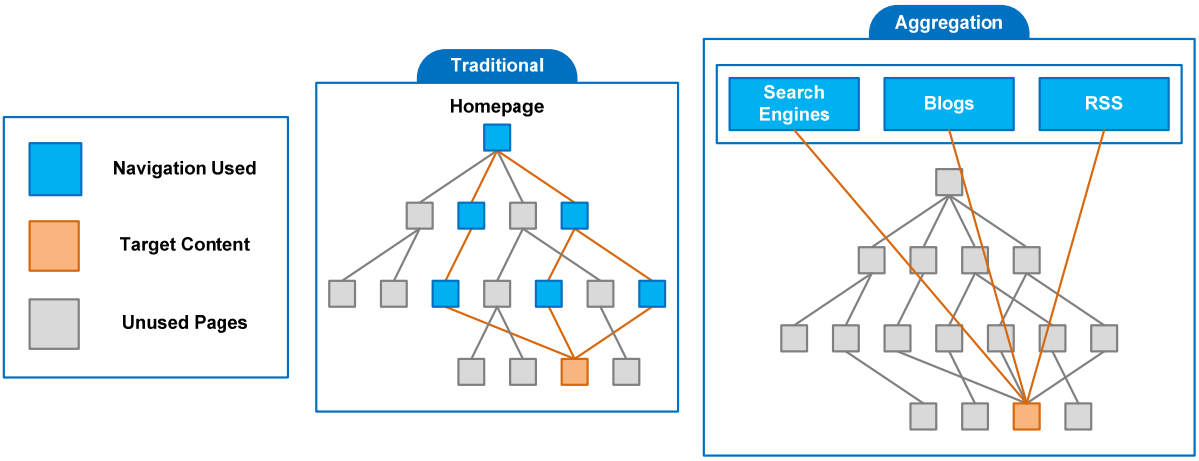


Figure 2-3 WWW Navigation Methods

Blog is a contraction of the term "Web Log" and refers to a web site which facilitates the easy publication of content, in publically displayed, time-ordered lists by a non-technical user. Most blogs provide regular commentary and personal opinion from the blogger on a specific subject or area. Technorati is a website which collects, organises and distributes information in relation to blogs. It essentially acts as a search engine for what has come to be termed "the blogosphere". In a 2008 study [Technorati 08], Technorati stated that 133 million blogs had been indexed since 2002. Of these, 7.4 million had been updated in the last 120 days. 1.5 million had been updated within the last 7 days and 900,000 had been updated within 24 hours. That constitutes a massive amount of user-generated content being created, updated and repurposed on a daily basis.

Blogs have potential for use in educational scenarios. A group of bloggers, publishing posts and comments on their blogs in relation to a subject area, can accumulate a corpus of interrelated knowledge. This group of bloggers could be learners in a classroom situation encouraged to collaborate by the educator, or independent learners interacting as they learn [Franklin & van Harmelen 07].

A wiki is a collaborative content creation tool which enables groups of users to generate, modify and contribute to pages of content. Wikis facilitate the construction of a corpus of

knowledge contained within a set of interlinked pages [Franklin & van Harmelen 07]. Wikis are often implemented in educational settings as knowledge management tools. Wikis are particularly suited to collaborative activities and can be implemented to monitor the incremental accumulation of knowledge by a group of learners.

The first wiki, WikiWikiWeb, was created in 1994 by Ward Cunningham as an attempt to make the exchange of ideas between programmers easier. Since then the number of wikis available on the web has grown enormously. The most widely known instance of a wiki is the web-based encyclopaedia Wikipedia [Wikipedia]. Wikipedia claims to be the world's largest multi-lingual encyclopaedia on the WWW. It currently has more than 7 million collaboratively edited articles in over 200 languages. These numbers are growing almost daily. Wikiversity [Wikiversity] is a Wikimedia Foundation project which attempts to collect learning content, from all levels and domains of education, and make it available for reuse under a variety of licences. Next generation semantic wiki's enable the formalisation and description of relationships between concepts and resources on a wiki [Tiropanis et al. 09]. This conceptual model of the knowledge contained in the wiki can be reasoned across to aid information retrieval and extraction. Examples of such semantic wikis are AceWiki [AceWiki] and KiWi [KiWi].

More formal collaborative content creation and editing tools have also begun to emerge. Examples of these platforms are Google Docs [Google Docs], Gliffy [Gliffy] and OpenGoo [OpenGoo]. Google Docs enables the collaborative creation and editing of content using online word processor and spreadsheet applications. OpenGoo is an open source solution which allows organisations to create, collaborate, share and publish all their internal and external content. It includes tools such as a calendar, task manager and word processor. Gliffy is a free web-based diagram editor. It allows users to create and share flowcharts, network diagrams, floorplans and other drawings. Such formal collaboration tools could be used in classroom group-work and assignment scenarios. Learners could simultaneously edit and collaborate on documents or diagrams.

As "web 2.0" applications have enabled all users of the WWW to become publishers of content, so mashups have stimulated web development by enabling anyone to combine existing data streams and develop web applications [Kulathuramaiyer 07]. Mashups are web-

based applications which seamlessly combine content from various sources and deliver an integrated user experience. While blogs and wikis promote the creation of text based content, mashups promote the user-driven design and development of new web-based applications. At the latest count there are approximately 3406 available on the WWW, this number is increasing at a rate of 3 mashups a day [ProgrammableWeb 08].

One of the main reasons why mashups are not being developed at a similar rate to blogs has been the lack of design tools and interfaces which can be used by the general, non-technical web user. These tools are now beginning to emerge, and as they grow in usability and popularity, so will the rate of mashups being generated. Examples of such design tools are Yahoo! Pipes [Yahoo! Pipes] and Microsoft Popfly [Microsoft Popfly].

The Extensible Markup Language (XML) is the foundation upon which much of the emerging content sharing and aggregation technologies have been developed. XML is used to encode and serialise content. Essentially, XML transforms traditional monolithic blocks of content into rapidly reconfigurable and reusable content modules. As its name suggests, XML is designed to be extensible and allows custom markup languages to be developed using it as a foundation. Through the addition of domain specific semantic constraints, many semantic markup languages have been created. These include MathML, MusicXML and the Chemical Markup Language (CML). These languages are helping to extend the portability and interoperability of content beyond the realm of natural language web-based text.

RSS, or Really Simple Syndication, is an XML-based standard which facilitates the aggregation of content from disparate and arbitrary sources on the WWW. iGoogle [iGoogle] and My Yahoo! [My Yahoo!] are two applications based on RSS which allow the individual to generate a personal homepage containing an aggregation of RSS feeds on their topics of interest. These feeds can be suggested and added automatically by the application or manually added by the user.

RSS was once referred to as a “killer app for education” [Harrsch 03] due to its ability to foster online communities of practice around specific areas of interest and its information subscription capabilities. RSS makes it possible for an educator or a learner to subscribe to a selection of known authoritative websites related to a topic being taught or learned. As soon

as content is updated on any of the sites, the user's feed is updated. They no longer have to check each source of information individually. When combined with blogging technologies, RSS could also be used in educational scenarios by formal class groups or informal groups of independent learners to track interaction between the educator and the group and between members of the group.

Social networking platforms enable the creation of communities or groups of individuals who can interact and share information. Examples are LinkedIn [LinkedIn], which facilitates professional networking, and Facebook [Facebook], which is an informal social site. Elgg [Elgg] is an open source social networking engine which enables the easy generation of web-based social networking applications. Indicators of ranking within communities can be integrated into social networking applications. This provides the opportunity to rank individual members of a community based upon their interactions and input. This acts as a motivator for the learner to contribute and become a valued member of a community, which reflects the situative approach to pedagogy discussed in section 2.2.

2.3.3.1 Analysis

The sheer volume of this web-based, user-generated content and its potential value in educational scenarios is beginning to be recognised within the TEL movement, and the limitations of TEL systems which cannot utilise such content are being exposed. Initiatives are being established, not only to try and capture this knowledge as it is produced on the WWW, but also to use the emerging web 2.0 tools to share verified educational content created within the academic domain. Attempts are being made to standardise content formats within the community so that content can be reused regardless of the TEL application being used as discussed in section 2.3.1. The design of TEL applications is evolving in an attempt to leverage this newly accessible wealth of information and knowledge.

However, despite the emergence of the content publication tools and socially oriented applications discussed in the above section, and the growth of communities of practice which use such tools to interact and collaborate with peers, it is not yet clear if these initiatives can be used to positively affect the formal sharing of educational content [Charlesworth et al. 07]. A recent JISC study [Margaryan 06] of attitudes and practices of educational content sharing found that only 9.7% of respondents used collaborative platforms such as websites, wikis and blogs for sharing work in progress. The forces which motivate interactions with like-minded

people in social contexts are not necessarily transferrable to more formal domains such as education. It is important to remember that the success and growth of these new interactive platforms is not merely a result of the emergence of new technologies, but a direct result of viral user engagement and participation. The real value of these platforms lies in supporting TEL applications through facilitating peer interaction and collaboration. It should be noted that social applications are known to be beneficial in constructivist and constructionist approaches to education.

There are also institutional concerns with regard to the use of web-based social and collaborative applications in education. Traditionally Universities have implemented institution-wide policies with regard to approaches to service and content provision [Stanley 06]. Course or even individual implementations of collaborative content technologies produce synchronisation and integration issues for institutions. There is a risk of producing a number of isolated islands of incompatible content. Institutions have also expressed concern about the longevity of such services and the resulting implications should they be implemented in formal educational scenarios. Services could be terminated without warning or move to a commercial model with charges associated. This could lead to the loss of service, or worse again, content, within the institution [Franklin & van Harmelen 07].

As digital content repository initiatives accumulate educational resources and web-based publication continues to produce enormous volumes of content, there remains no centralised method of collating and utilising content, related to a particular subject domain, from all these various sources. More and more content is being placed online and available for use but the discoverability and aggregation of content from disparate sources remains an issue. RSS is an excellent tool for notification of newly published content and the aggregation of related content, however the user must be aware of authoritative sources of this information and actively subscribe in advance.

2.3.4 Content Utilisation in Personalised TEL

Technology enhanced learning systems which facilitate adaptivity and personalisation[6] have traditionally operated upon closed sets of content resources, where the system is aware of

[6] Adaptivity is a broad term used to describe dynamic changes to the system interface or content displayed based upon certain influencing parameters. Personalisation typically refers to a particular type of adaptivity, namely

each individual resource, and any relationships between resources, in advance [Aroyo et al. 03]. These content resources are typically authored for use in a specific TEL environment, conforming to a proprietary structure and as a result are not easily portable or reusable. Such systems are thus referred to as closed corpus in nature.

Intelligent Tutoring Systems (ITS) [Urban-Lurain 96] were one of the first initiatives to attempt to facilitate the generation of personalised TEL experiences. However, the implementation of functionality to support personalisation resulted in adaptivity sequencing logic and navigational controls being embedded in the actual educational content [Conlan 05]. This situation is far from ideal as this content cannot be subsequently reused outside of its original operating context or environment. The incorporation of externally authored content or alternately structured content is also infeasible as a result of this implicit technical co-dependence between the content format, adaptivity information and TEL system. Examples of such ITSs include ELM-ART [Brusilovsky et al. 96], Interbook [Brusilovsky et al. 98] and early versions of AHA! [De Bra & Calvi, 98].

Adaptive Hypermedia (AH) combines concepts from both the ITS and WWW hypermedia communities. Adaptive Educational Hypermedia Systems (AEHS) utilise learner and domain models to achieve the adaptive selection and sequencing of linked hypermedia content for the learner [Brusilovsky 98]. Early AEHSs adopted a similar approach to ITS by embedding adaptivity sequencing logic and navigational controls in the educational content used by the system. Recently developed generations of AEHS have attempted to support dynamic functionality such as adaptivity and personalisation. The content requirements of these systems are more stringent and the volumes of content consumed are far greater than non-adaptive TEL systems. As a result, these systems are being hampered in efforts to deliver such dynamic functionality by their traditional closed corpus nature and reliance upon bespoke, proprietary content. To scalably support such system flexibility and dynamism in mainstream education, TEL applications require access to large volumes of educational content which is varied in structure, language, presentation style, etc [Brusilovsky & Henze 07].

---

adapting the system with a focus on learner characteristics and their environment. In this thesis, although these phrases may be frequently interchanged, unless explicitly stated, the focus is on adaptive personalisation.

In an attempt to address the functionality and scalability issues described above, a "second generation" of AEHSs have emerged. These systems have instigated a significant shift towards the detachment of adaptivity controls relating to content sequencing, and the physical educational content. This development, in principle, enables the integration of externally developed, open corpus, content into these previously closed corpus systems. However, due to proprietary restrictions enforced by such systems on content format and structure, the majority still source all educational content from system-exclusive repositories of learning resources. Examples of such systems which typically require the apriori development of bespoke learning resources are Knowledge Tree [Brusilovsky 04b], AHA! [DeBra et al. 03] and APeLS [Conlan & Wade 04]. The ability of AH applications to function using content from open corpus sources has been questioned in recent times, and has been labelled the "Open Corpus Problem" [Brusilovsky & Henze 07]. AEHS which attempt to reap the benefits of educational content from sources such as digital content repositories, LOs and the WWW must be able to function without placing complex structural restrictions on content and without the need for design-time knowledge of available resources and relationships between these resources.

Some educational hypermedia systems, such as KBS Hyperbook [Henze & Nejdl 00] and SIGUE [Carmona et al. 02], allow the incorporation of content sourced from the WWW. However, leveraging such content for use within these systems requires significant manual effort on the part of a domain expert or course designer, in advance of incorporation. The content must be sourced, not a trivial task, and then tagged using the LOM [LOM] metadata schema and associated with domain model concepts to enable integration [Henze & Nejdl 01]. SIGUE, similarly, requires the metadata generation for all new content and manual integration with a domain model. Knowledge Sea II also allows the integration of open corpus content. To add an open corpus resource to the system, a comparative analysis must be conducted between the new resource and every existing resource in the collection. A keyword-based similarity metric is used. Knowledge Sea II employs self organising topic maps to categorize content and create a linked flow between resources [Brusilovsky et al. 04]. The open corpus resources integrated by the system were specifically selected, hierarchically structured web-based textbooks and not general content from individual web pages.

2.3.4.1 Analysis

The current necessity for the apriori development of educational resources for use in TEL applications creates a situation whereby educators who wish to utilise such technologies and approaches are forced to author large volumes of high quality educational content specific to each educational scenario. TEL content creation is expensive [Marchionini 95] [Boyle 03] and the authoring of content across broad subject spectrums, structural varieties and presentation styles can quickly become an extremely extensive undertaking.

Content-based recommendation systems have been used to successfully recommend open corpus content that is relevant to a user's interests [Pizzani & Billsus 07]. However these systems can struggle with low precision in the quality of the resources they recommend, due to the inconsistent nature of the WWW. Apriori content classification and filtering on the pool of potential open corpus resources available to such recommender systems would greatly improve the quality and accuracy of the resources delivered to the user. By ensuring that only highly relevant educational material is presented to the learner, information overload can be greatly reduced.

The first step to achieving progress in developing open corpus TEL systems that are comparable in functionality to closed corpus systems is to devise a method of discovering and collating large volumes of good quality, domain-specific open corpus content and making this content accessible to the system. Once content has been identified and harnessed, research can be undertaken into the addition of knowledge-driven semantics to the content [Berners-Lee et al. 01], such as semantic annotation and content linking. These semantics can be used in a similar fashion to the personalisation controls in closed corpus systems, to facilitate adaptivity.

2.3.5 Digital Rights Management and Intellectual Property

Arguably the most significant barrier to engagement with digital content repositories, and the adoption of educational content sharing initiatives such as LOs, is uncertainties and misconceptions with relation to copyright law, ownership and intellectual property (IP) [Casey et al. 07]. There appears to be a significant degree of confusion with regard to the ownership of educational materials and research materials created by academics [Davis et al.

09]. When academics were given the question "who owns the copyright on teaching materials" in a recent survey, 54.9% responded that they were unsure [Bates et al. 07].

Typically the copyright for educational material used for teaching belongs to the institution whereas the copyright for content relating to research, particularly journal papers, typically resides with the publishing body. In a 2003 study, 32% of content authors questioned were unaware of the copyright status of their journal articles [Gadd et al. 03]. Academics have resultantly become wary of sharing content because they don't know if they can or because it involves a complex sign-off process. An additional concern, stemming from the fact that Intellectual Property Rights (IPR) with relation to published content is still perceived as unclear, is that many researchers may fear that publication of research material could impede upon the future commercialisation of that research.

The emergence of collaborative web-based content authoring tools and social applications only serves to exacerbate the difficulties in properly protecting content. Educational content generated using such tools will typically have been authored by numerous individuals, often in different countries or jurisdictions, who may not even know each other [Franklin & van Harmelen 07]. Attempting to enforce copyright and ownership in such circumstances is a non-trivial task. In an educational scenario this issue can be further complicated if the collaborative system is hosted externally. There can be additional IPR issues if external collaborators can contribute to content generated by the system.

The development of copyright management methods such as Open Source licenses [GNU] for reusable software and Creative Commons licenses [Creative Commons] for the protection of artistic works and content have been positive steps in relation to information sharing. However, there has yet to be widespread adoption and implementation of such standards, particularly in the education domain. This is largely a result of confusion between licenses, and their implications for dissemination and exploitation strategies, at both an institutional and individual level [Charlesworth et al. 07].

2.3.6 Summary

This section discussed the current trends and approaches in the authoring and distribution of digital content with particular reference to content in the educational domain. The section also described the current methods of educational content utilisation in TEL applications. The

WWW has revolutionised the means by which information can be created and shared. Content can now be collaboratively authored and instantly published to the WWW. This means that it can be potentially shared with millions of individuals worldwide at essentially no cost. The value and scale of user-generated and collaboratively authored web-based content is beginning to be recognised. Changes in the design of TEL applications have begun to emerge, as they attempt to utilise this new wealth of openly accessible educational content. This creates the potential for content reuse across TEL implementations and educational contexts. Digital content repositories, content publication applications, social platforms and collaborative applications all facilitate, to some level, the dissemination of content. However, the repurposing and reuse of that content outside of its original context remains a difficult and challenging task.

2.4 Summary

This chapter examined the current trends and issues with the creation of educational content for use in technology enhanced learning. Section 2.2 detailed a categorisation of learning and the associated pedagogical approaches. A mapping of these broad and overlapping perspectives into the design of pedagogically effective TEL applications was also explained. Section 2.3 discussed the recent changes in the creation, management and dissemination of digital content. This was discussed with a specific focus on educational content and its utilisation within TEL systems. The chapter concluded with an analysis of the evolving nature of educational content reuse in TEL.

3 State of the Art of Information Retrieval on the WWW

3.1 Introduction

As educators attempt to incorporate TEL offerings into traditional course curricula, the lack of availability of appropriate digital content resources acts as a barrier to adoption [Brusilovsky & Henze 07]. The development of quality TEL resources has proven to be an expensive undertaking [Boyle 03], yet to scalably support next generation functionality such as personalisation and on-demand adaptivity access to large volumes of varied educational content is required. As discussed in chapter 2, the potential value of the vast quantities of content available via the WWW is beginning to be recognised within the TEL movement. Numerous digital content repository initiatives are beginning to accumulate educational resources and web-based publication now produces enormous volumes of content. Much of this content could be used by TEL applications within educational scenarios. However, within the majority of individual digital content repository initiatives there remain insufficient volumes of contributed content to spawn regular and widespread content reuse.

As the web grows it will become increasingly difficult for educators to discover and aggregate collections of relevant and useful educational content [Ring & MacLeod 01]. There is, as yet, no centralised method of discovering, aggregating and utilising educational content, from these various disparate sources, within TEL. This research proposes that Information Retrieval (IR) techniques and technologies could be applied to traverse the WWW and centrally collate educational resources, categorised by subject area. These subject specific collections of content could then be used by TEL applications during the generation and execution of learning experiences.

The objective of this chapter is to provide the reader with a detailed insight into the foundations and current trends of Information Retrieval on the WWW. This chapter is used to identify the appropriate techniques and technologies required to support the discovery, classification and harvesting of educational content from open corpus sources. An introduction is provided to the evolution of Information Retrieval and the development of the WWW. The chapter then provides a detailed description of web crawling and web-based IR algorithms which support content discovery on the WWW. The growth of the WWW is then examined. This acts as an introduction to, and foundation for, the analysis of past and current

focused crawling techniques and systems which enable the subject-specific discovery of content. Content indexing and retrieval are then discussed in relation to web-based content. Based on the analysis in each section, the chapter concludes by discussing the elements of web-based IR which could be used to influence the discovery and collation of subject-specific educational content for use in TEL.

3.2 The Evolution of Information Retrieval

Information retrieval (IR) is a scientific field which evolved in response to the various challenges of locating and accessing relevant, or sought after, information. IR proposed and developed a principled approach to searching for documents, for information within documents and for metadata describing documents. IR systems are generally concerned with receiving a user's information need in textual form and finding relevant documents which satisfy that need from a specific collection of documents. Typically, the information need is expressed as a combination of keywords and a set of constraints. Most IR systems have focused on storing and viewing documents, methods for processing queries and determining document relevance, and user interfaces for querying and refining results [Sampath 1985].

Over the last two decades, with the emergence of the WWW, the field of IR has evolved. What was once a discipline primarily applied in academia, now forms the foundations which underlie most mainstream means of sourcing and accessing information. Conventional IR systems traditionally catered for experienced users of the system and domain experts such as librarians and academics. However, as the WWW gained more popularity and mainstream use, web-based IR systems could not make assumptions regarding the users' level of knowledge with regard to the system or the subject domain. The IR system has to cope with users across the spectrum of knowledge and ability [Hölscher & Strube 00].

The two metrics generally used in the evaluation of IR systems are *precision* and *recall* [Salton 89]. Precision is the fraction of documents retrieved in response to a query that are relevant to the user's information need when making that query. Recall is the fraction of the total set of relevant documents in a collection which are returned for a given query. These metrics are highly subjective, as relevancy can only be assigned based upon the user's intent when submitting a specific query. The structure of the document collection alters the metrics which can be used to assess the performance of IR systems. In relation to the WWW, recall is

an ineffective means of evaluation, as recall requires the entire collection of relevant documents to be known in advance of each query performed. This calculation cannot be performed on web-based collections as the entire set of relevant pages is unknown. The WWW is constantly growing and evolving, and there is currently no complete index of the entire web in existence.

3.3 The World Wide Web and Hypertext

A hypertext, originally conceived by Vannevar Bush [Bush 45] and later coined by Ted Nelson [Nelson 81], can be defined as a collection of information fragments, or nodes, that have active cross-references known as hyperlinks. These links allow an individual browsing a node in the collection to jump to another node in a different location when desired [De Bra et al. 94]. This loosely coupled structural design means that information, or new nodes, can be added to the collection at any point and at any time. However, the majority of early hypertext systems were constructed using individual collections of documents, often in a single subject domain [Coombs 90].

The WWW [Berners-Lee et al. 92] is essentially a distributed hypertext on a massive scale. Nodes can potentially be located anywhere in the world, contributed by millions of authors and consumed by even more readers, as described in more detail in section 2.3.3. However, the drawback to this ease of publication is that there is no organised method to catalogue or list the nodes contained in the collection. The Hypertext Transfer Protocol (HTTP) is used to retrieve individual nodes from the server on which they reside, but there is no protocol for discerning what servers are in existence and what nodes are on each server [De Bra & Post 94a].

By unleashing publication on such an unprecedented scale, the WWW has acted as a principal driver of innovation in the IR field. This explosion of published information would be moot if information, relevant to a user's interests and needs, could not be easily and quickly located. Web Search Engines were to become the dominant IR mechanism on the WWW. These search engines are services which are largely fed by web-traversing robots. By exploiting the hypertextual nature of the WWW, these robots browse nodes on the web, recursively following the hyperlinks contained in each node, archiving content and constructing a database of the pages encountered. These robots later become known by

various pseudonyms such as web spiders, web crawlers, web robots and web scanners[7]. The following section will examine web crawling and the underlying link analysis algorithms.

3.4 Web Crawling and Web Search

Traditional search engine architectures have three main components: the web crawler, which traverses the web in an attempt to discover new or modified content; the indexer which creates a searchable reference of all the content encountered; and the user interface which allows the individual to express their information need and filter through the content returned. Each of these components will be examined in turn over the following sections. However, to fully understand how these web-based IR services have evolved and the means by which they function, it was necessary to examine their emergence and development over the past 15 years. This review can be found in Appendix B.

The development of two algorithms which exploit the hyperlinked structure of the WWW, were key to the emergence of recent web crawling and web search techniques. These algorithms allowed Google [Google] to gain market dominance over its rivals, despite its late arrival upon the web-based IR scene. An examination of these algorithms will give some insight into the structure of the WWW and how this structure can be exploited when conducting focused crawls for particular subject areas and domains, as will be discussed in section 3.6.

3.4.1 Web-based IR Algorithms

The practice of link analysis on the WWW has its foundations in the area of bibliometrics [Pritchard 69]. Citation analysis is one of the most commonly used bibliometric methods. It involves the study and analysis of citations in published literature to produce quantitative estimates of the importance and impact of individual scientific papers and journals [Kleinberg 99b]. Citations have been described as “frozen footprints on the landscape of scholarly achievement” [Cronin 84]. The most well-known measure in this field is Garfield's impact factor [Garfield 72], used to provide a numerical assessment of journals. The impact factor of a selected journal is calculated as the average number of citations received by papers published in that journal over the previous two years [Egghe & Rousseau 90].

[7] In this thesis such archival *robots* will, from this point onwards, be referred to as crawlers.

Pinski and Narin [Pinski & Narin 76] proposed a more subtle citation-based measure of publication impact. This improved measure is based upon the observation that not all citations are of equal importance. In this approach, a journal can be deemed influential if it is heavily cited by other influential journals. Put simply, a citation from an influential, high-profile journal should carry more weight in the generation of a journal's impact factor.

3.4.1.1 Hubs and Authorities

Jon Kleinberg noted the parallels between the hyperlinked structure of the WWW and the bibliographical structure of academic publications [Kleinberg 99b]. Kleinberg defines the WWW as "an intricate form of populist hypermedia, in which millions of participants, with diverse and often conflicting goals, are continuously creating hyperlinked content". The application of bibliometric methods to the study of the WWW is termed webometrics, and is defined as "the study of the quantitative aspects of the construction and use of information resources, structures and technologies on the WWW drawing on bibliometric and informetric approaches" [Almind & Ingwersen 97] [Björneborn & Ingwersen 04].

The analogies between the citation structure of scholarly journals and the WWW are immediately obvious. The creation of a hyperlink between two pages, *A* and *B*, on the WWW can be used to infer the following: The author of page *A*, by creating a link to page *B*, has conferred some measure of confidence and authority on *B*. The author of *A* is stating that they believe *B* to be a relevant and reputable resource in relation to the subject under examination. Pinski and Narin's extension to citation analysis can also be applied to hyperlink analysis on the WWW. A hyperlink to a page is more important if it comes from a highly respected, influential web-page. Links to a page *A* from the BBC, CNN and Yahoo! homepages potentially make *A* more important that a page *B* which is linked to from 100 small, obscure websites. This pattern holds true, particularly in non-commercial environments, such as educational or informational web pages where high quality resources tend to link to other authoritative sources of information on that subject. The pattern differs in commercial environments where it is common for authorities not to link to the other main authorities in their area, as they are often in direct competition with these websites. This theory of web-graph structure is known as "Hubs and Authorities" [Kleinberg 99a].

Kleinberg identifies a diversity of roles among web pages in a common subject area. Pages which have a large volume of incoming links, particularly from other influential pages, prove

to be the most prominent sources of information on a particular subject. These pages are labelled “authorities”. Other pages, which are equally intrinsic to the structure of the WWW, offer collections of links to numerous recommended, high quality sites, or authorities, in a particular subject area. These pages, labelled “hubs”, act as reputable resource lists. The nature of the relationship between hubs and authorities is very asymmetrical. Hubs link heavily to authorities but may themselves have very few incoming links. This theory of web-graph structure is illustrated in Figure 3-1 below.

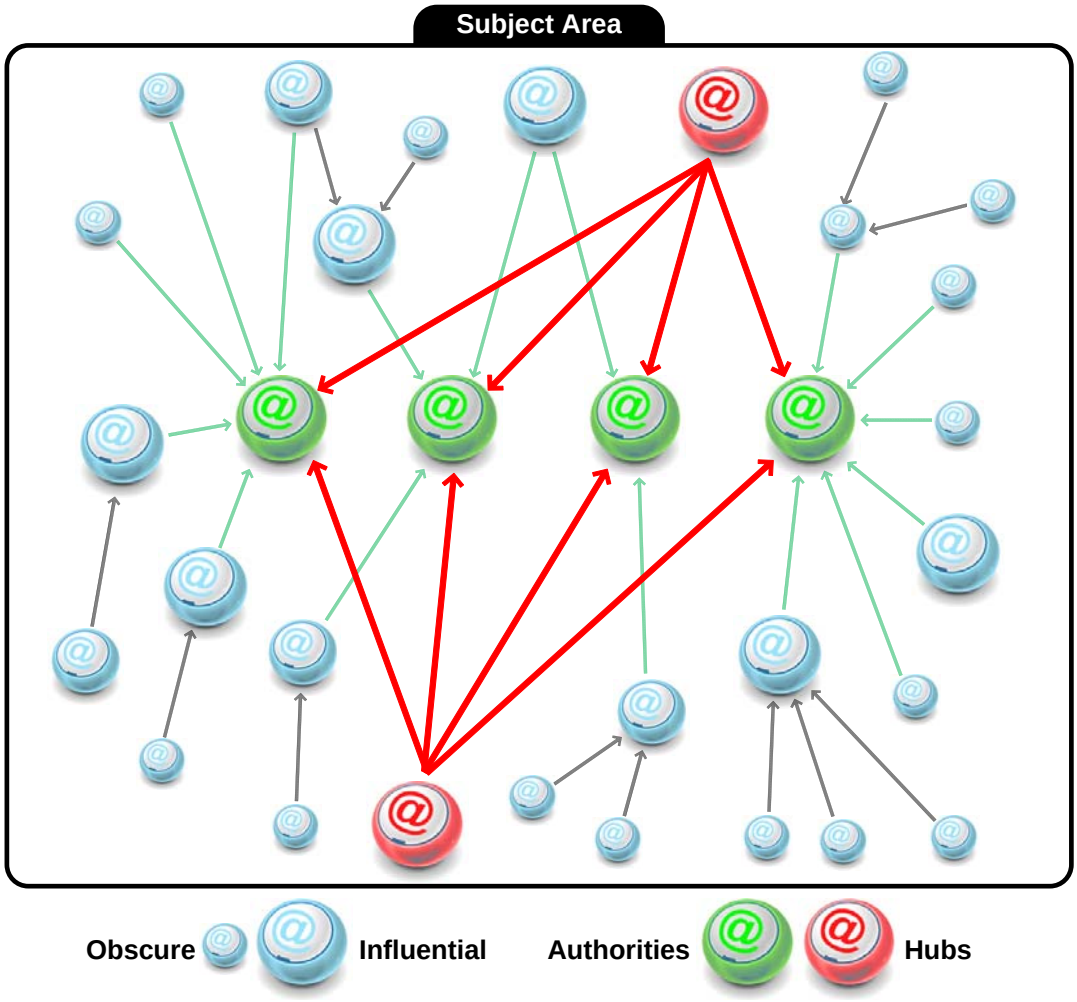


Figure 3-1 Hubs and Authorities.

If page A links to page B, it can be inferred that A is recommending B. If A is linked to by a large number of influential documents in a subject domain, then A is an authority on that subject. If B links to a large number of influential documents in a subject domain, then B is a hub for that subject domain. A good authority will be linked to by many good hubs, and a good subject hub will link to many good authorities.

3.4.1.2 HITS

Hyperlink-Induced Topic Search (HITS) is a ranking algorithm developed using the hubs and authorities theory. In response to an IR query a text based search is invoked. HITS is then applied to a small sub-graph of the web constructed from the search results. A hub weight and authority weight are calculated for each node in the sub-graph. A node's authority weight is proportional to the sum of the hub weights of nodes *which link to it*. A node's hub weight is proportional to the sum of the authority weights of nodes *which it links to*.

3.4.1.3 PageRank

PageRank is another algorithm based upon these structural patterns. It was developed by Larry Page and Sergey Brin at Stanford University between 1995 and 1998 and went on to form the basis of the Google [Google] search engine. PageRank was developed as an attempt to calculate a global ranking of every page on the WWW, regardless of its content, based solely on the page's location in the Web's graph structure [Page et al. 98]. The algorithm extends the notion of "authorities" in Kleinberg's work. PageRank interprets a hyperlink from page *A* to page *B* as a vote by page *A* for page *B*. The algorithm also analyses the page from which the hyperlink has emanated. Links from pages that are deemed important in their own right are more heavily weighted and have more influence on the resulting rank calculated for the linked to page. More details on the PageRank algorithm can be found in Appendix B.

3.4.2 Analysis

The main disadvantage of link analysis algorithms such as HITS and PageRank is that they favour older pages. New pages, even if their content is of very high quality, will initially not have many incoming links unless the page is part of an existing site's link structure. Various methods of manipulating link analysis algorithms have been attempted in an effort by commercial sites to improve search results rankings and monetise advertising links. These strategies have impacted the effectiveness of such algorithms when used in isolation. Most search engines use a combination of factors such as traffic volumes and click through rates in combination with these algorithms. Search engines are also known to actively penalise sites designed to artificially inflate their ranking. In December 2007 Google began actively penalising sites selling paid text links [Link Buying].

Although link analysis algorithms are primarily used as a ranking methodology for the serving of search results, they can also have implications upon the way the web is crawled for information. The structure of the web described by these algorithms can be exploited when crawling the web for content in a particular subject area. This directly influences the design of an application which supports the discovery, classification, harvesting and delivery of topic-specific educational content from open corpus sources, which is a goal of this research. The exploitation of link structure in focused web crawling will be discussed in more detail in section 3.6.

3.4.3 Summary

The previous two sections described web crawling and web search methodologies and the link analysis algorithms upon which they are based. The hypertext structure of the WWW reflects bibliometric patterns, and can be compared to that of a scholarly journal. Quality resources tend to link to other authoritative sources in the same subject area, much like a reference in a scholarly journal paper. This pattern is detailed in the Hubs and Authorities theory and exploited by algorithms such as HITS and PageRank. Exploiting this structural pattern and implementing IR techniques based on bibliometrics could be extremely useful for discovering quality sources of subject-specific content.

3.5 WWW Growth and the Deep Web

The publishing and dissemination of content on the WWW is becoming much easier and more accessible, as discussed in section 2.2.3. This has resulted in enormous growth in the volume of digital content available for consumption. As the growth of the web continues to accelerate, tracking the emergence of new information, and sourcing specific information within this distributed network of nodes and servers becomes increasingly difficult. This section will introduce recent trends in the growth of the WWW, and how this growth has affected the ability of IR tools to perform efficiently and effectively. This section will also explain how this growth has promoted the development of a new form of web crawling and IR on the web called “focused crawling”.

3.5.1 WWW Growth

The growth of the WWW has been rapidly increasing, particularly since the turn of the century, and the exact size of the web has become a contentious issue. Two of the main publishers of statistical information regarding the WWW are Netcraft [Netcraft] and the

Internet Systems Consortium (ISC) [ISC]. Some estimates place the creation of new pages on the WWW at 320 million per week [Ntoulas et al. 04].

Netcraft publish statistics on the number of web servers detected each month. While Netcraft do not publicly state how they generate their figures, it is widely believed that they use a web crawler to traverse the web and send requests to each server encountered. In the Netcraft statistics for April 2008, they estimate that there are a total of 165,719,150 web servers on the WWW, of which approx 70 million are active. The ISC produce statistics which measure the number of individual hosts on the WWW. This is accomplished through a domain name survey, calculating the number of IP addresses that have been assigned. The ISC statistics for January 2008 reported 541,677,360 hosts on the WWW. The essential difference between the Netcraft and ISC metrics is that the ISC include both producers and consumers of information in their calculations, whereas Netcraft include only producers of information.

While the size of the WWW has been rapidly increasing, similarly has the number of people using this vast network. Internet World Stats [IWS] publish statistics on the number of people worldwide accessing the WWW. This has grown from an approximate figure of 16 million in December 1995 to over 1.3 billion today. This rapid growth of the WWW has been exacerbated by the recent explosion of web content publication beyond the traditional territories of North America and Europe, and beyond the traditional social strata of academia and industry.

The WWW user base conventionally had a defined split between producers and consumers of information. However, the ease with which information can now be published on the WWW and the relative prevalence of broadband access has resulted in the emergence of a new type of user, the pro-sumer. These users both produce and consume content. The steady growth of this type of user is directly linked to the proliferation of media sharing sites, blogs, social networking and, in the field of education, to digital content repositories, as previously discussed in section 2.2. For more detail on the growth of the WWW see Appendix B.

3.5.2 The Deep Web

The loosely coupled design of the WWW means that some nodes on the network may have no incoming links and thus are essentially “undiscoverable” by conventional web crawlers. This portion of the web, unreachable when employing link analysis algorithms, has become

known as the “Deep Web”. In 2001 it was estimated that public information on the deep web was between 400 and 550 times larger than the mainstream WWW [Bergman 01].

One of the reasons a page may have no incoming links is if the content is dynamically generated. An example of this is commercial web pages which allow the user to submit a query in order to retrieve a list of products. The required page is then dynamically constructed, usually by querying an internal database of potential information. Various web crawling techniques have been developed in an attempt to access such dynamic content. One approach is to generate and submit sets of keywords to site query boxes to generate and retrieve most of the potential dynamically generated pages [Raghavan & Garcia-Molina 01] [De Carvalho Fontes & Silva 04] [Ntoulas et al. 05]. A second approach is to simulate an individual’s browsing behaviour on the WWW using agents [Lage et al. 04].

In a study on the hypertext connectivity of the WWW [Broder et al. 00], web pages were categorised based upon their link structure. Three main categories and two sub-categories were defined. The analysis was conducted using over 200 million individual pages and approximately 1.5 billion hyperlinks collected by a web crawl in 1999. This approach to WWW link analysis has become known as “BowTie Graphy Theory”, for reasons which become obvious upon viewing the graphical representation produced by the study, reproduced in Figure 3-2 below.

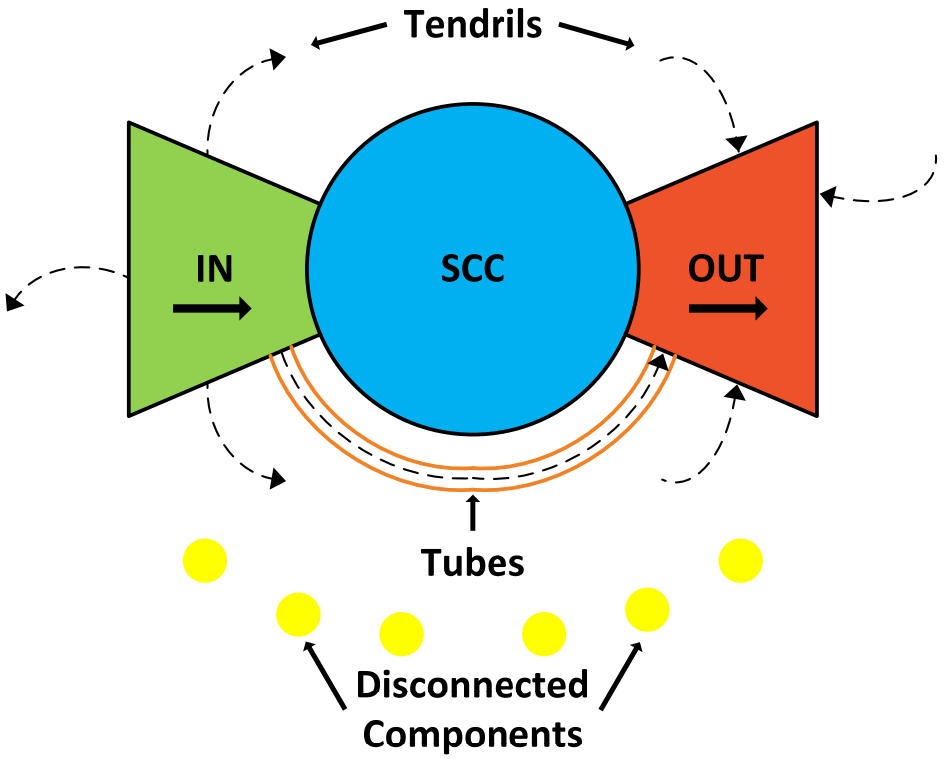


Figure 3-2 The BowTie Graph Theory [Broder et al. 00]

The Strongly Connected Component (*SCC*) category consists of pages that can reach one another along directed links. The second and third categories are called *IN* and *OUT*. IN consists of any pages that can reach SCC pages but cannot be reached by any SCC pages. Out consists of pages that can be reached by SCC pages, but do not link back to any SCC pages respectively. The rest of the pages, called Disconnected Components, cannot be reached and do not reach any SCC pages. In theory an individual or web crawler browsing the WWW can pass from any page within the IN category, through SCC to any page in OUT. Hanging off IN and OUT are Tendrils. These Tendrils contain pages that are reachable from other pages within IN, or that can reach portions of OUT, but which do not link to SCC. It is possible for a Tendril page from IN to link to a Tendril page which leads into OUT, forming a Tube.

Of the 200 million web pages analysed [Broder et al. 00] over 56 million, or 27%, were categorised as strongly connected and were placed in the SCC. 43 million, or 21%, were found to link into the SCC set but were not linked to by the SCC set and such were placed in IN. Similarly, 43 million, or 21%, were linked to by pages in the SCC set but did not link back to any pages in the SCC set and were placed in OUT. Another 43 million were found to link to other pages within their category but never to the SCC and were placed in Tendrils.

Finally 16 million pages were found to be Disconnected Components. This graphical description of the WWW explains why it can be difficult for web crawlers to discover and access significant portions of the web. If the crawler starts its navigation from a page in the OUT set, it can only reach approximately one fifth of whole WWW.

All the information that a search engine is unable to access has been termed "Dark Matter" [Bailey et al. 00]. This not only includes the deep web, but also pages which are intentionally hidden from web crawlers through the Robots Exclusion Protocol [Robots.txt]. The Robot Exclusion Protocol [Robots.txt], also known as the Robots Exclusion Standard or robots.txt, was established as an independent method of defining web crawler "politeness" by the members of the robots mailing list, robots-request@nexor.co.uk, in June 1994. The protocol is a convention by which cooperative, or polite, web crawlers can be requested not to access all, or parts, of a website which is otherwise publicly accessible. It is at the discretion of each crawler whether they obey the this protocol. However, crawlers which disregard it are often targeted by spider traps or have their IP addresses blocked entirely by web servers.

3.5.3 Analysis

The overwhelming challenge of attempting to source specific content on the ever expanding WWW is, in part, due to a traditional one-size-fits-all philosophy. Google and many similar search engines which utilise web crawlers attempt to cater for every possible search that may be performed. Although such services are invaluable due to the broad spectrum of information they cover, the resulting diversity of content often results in thousands of responses of little relevance or quality to all but the most expertly constructed queries.

As a result of this volume and diversity of information, the individual can suffer from information overload as they attempt to comprehend the expanse of unstructured material being presented. They may become disoriented, unable to determine how the current information displayed relates to the information they seek [Eklund 95]. This phenomenon has become known as 'Lost in Hyperspace' [Conklin 87] [Edwards & Hardman 89], and is a frequent occurrence in situations where people are exposed to large volumes of apparently disorganised information. Even experienced users can become spatially confused when presented with information whose structure is seemingly random and incoherent [Laurillard 93] [Theng 97].

This becomes an even more acute problem when examined within the scope of education. In learner-driven educational experiences, it can often be extremely difficult for the learner to filter the information which they encounter. This is the case not only with regard to the relevancy and applicability of new information, but also with regard to the quality of the on-topic information. It is not sufficient in an educational experience to provide a student with a mixture of on and off topic content, of high and poor quality content. Students can be distracted and become disengaged from the educational experience by encountering irrelevant information or due to frustration with the quality of content encountered.

More notice is now being given to the fact that the massive scale of traditional web IR solutions can also be their biggest disadvantage. There is a growing sense of natural limits within the web IR industry, a recognition that in certain circumstances covering a single galaxy can be more practical, and useful, than trying to cover the entire universe [Gillmor 98]. This realisation has resulted in the emergence of a new approach to web crawling, termed "focused crawling". This is a methodology whereby crawls are conducted for specific scenarios or subject areas using automatic filtering of the content encountered to estimate relevance. This can be used to generate smaller, more specific content collections which can be accessed through topical portals.

It is the aim of this research to collate and deliver quality, topic-specific collections of educational content to both the educator and learner within a technology enhanced learning environment. This content should be filtered to ensure that only content that is relevant to a specific subject domain or scenario is included in the pool. Focused crawling can provide both a means of content discovery and filtration for the creation of these content collections. The state of the art of focused crawling will be discussed in more detail in the following section.

3.5.4 Summary

This section discussed the recent surge in the growth of the WWW. The sheer volume of information available via the WWW makes it increasingly difficult to source relevant information. This has negatively affected the ability of web-based IR tools to perform efficiently and effectively. This section also introduced the concept of the deep web and the loosely coupled connectivity of the WWW. The following section aims to provide the reader

with a detailed introduction to focused crawling and the various techniques of employed to efficiently perform topic-specific content discovery.

3.6 Focused Crawling

The goal of a focused web crawler is to selectively seek out pages that are relevant to a pre-defined set of topics. A focused crawler can be defined as a web crawler which actively seeks, acquires, indexes and maintains pages on a specific topic which represent a relatively narrow segment of the WWW [Chakrabarti et al. 99]. Besides sourcing content based on its content, focused crawling allows a web crawler to process specific sites to greater depths than general purpose crawlers. Focused crawlers can spend more time perusing highly relevant sites rather than attempting to attain broad coverage of the entire WWW in a breadth-first manner. As a result, highly relevant pages can be discovered that may have been overlooked by more general-purpose crawlers [Chakrabarti et al. 99].

Generic web crawlers, and the search engines based upon them, can be compared to public libraries: they try to cater for the public at large, with a supply of material to meet the majority of common information needs and preferences. Such systems do not specialize in specific areas of interest. This creates a problem when attempting to apply such technologies within the education domain. In TEL, the content required for the generation of educational experiences is often highly specialised and must meet the requirements of both the educator and curriculum.

Consider a zoologist, highly experienced in her field, who delivers lectures in a very specific subject area, such as the Yangtze River Dolphin. The exponential growth of the web matters little to this educator if only 10 pages in her particular topic of interest are added or updated on a weekly basis. However, staying abreast of where this content resides, when and where new content appears and when content is updated is a non-trivial task which becomes increasingly difficult as the web expands. To locate such content, traditional web crawlers would waste huge amounts of computational resources traversing and indexing hundreds of millions of pages when the number of pages which are relevant and desired by the educator may be very small.

Now consider a very inexperienced learner who is only beginning to study a subject area. As discussed in the previous section, the WWW is vast in scale and offers information on a huge variety of topics, the boundaries between which can often be blurred. When searching for information, people attempt to avoid information overload by filtering the information with which they are presented both by relevance to their information need and by quality. For an inexperienced learner this can be a bewildering experience. They have very limited knowledge of the subject area in question and as such, their ability to judge relevance is impaired.

A system which can explore the WWW searching for sources of quality educational content on a particular subject and periodically re-crawl these sources for updates would be an invaluable tool. A suitably implemented focused crawling system could provide just such a service. The various methods employed by focused crawling systems to facilitate the discovery of topic-specific content on the WWW are discussed in detail in the sections below. However, before discussing individual approaches, Topical Locality, a theory which underlies almost all focused crawling techniques will be explained.

3.6.1 Topical Locality

Topical locality refers to the observation that web pages are typically linked to other pages with semantically similar content. In other words, web pages tend to link to other pages which contain related information. Focused crawlers exploit this phenomenon to aid the topic-oriented discovery of web pages.

In an examination of this theory [Davison 00], topical locality was combined with an analysis of the anchor text of hyperlinks as a discriminator of the relevance of unseen pages. Evaluations were conducted on a random selection of 100000 pages from the archive of the DiscoWeb search engine. The evaluation results showed that a page is significantly more likely to be topically related to the pages to which it is linked, as opposed to other nearby pages or randomly selected pages. Sibling pages are also more topically similar when the links on the parent are physically located close together. The results also found that anchor text is often very informative about the contents of the page it references, and as a result can be useful in discriminating among unseen child pages. Davison found that the inclusion of text in close proximity to the hyperlink did not significantly improve similarity measures.

Similar conclusions were drawn in an evaluation conducted in [Menczer 04]. Topical locality was formalised as two conjectures. The "link-content conjecture" states that a page is likely to be similar in subject matter to the pages that link to it. The "link-cluster conjecture" states that pages about the same topic are likely to be clustered together [Menczer 01], i.e. the content of a page can be inferred not only by examining the pages that link to it, but also by examining its neighbours. An experiment was conducted to validate these formalisations by analysing the correlation between lexical similarity and link distance. The evaluation showed that the lexical similarity of two pages exponentially decays as the range of links between the pages increases. The probability of a page being topically relevant is high within a radius of three links. It then decays rapidly. This indicates that the performance of a focused web crawler can be significantly affected by the proximity of clusters of relevant pages.

3.6.2 Methods of Focused Crawling

The first topic-specific or focused crawling technique dates back as far as 1994, when many of the early general purpose web crawlers were still emerging. However, it was to be upwards of five years before the next developments in the area appeared. Since then, many approaches to focused crawling have emerged, all of which use different techniques to aid the crawler as it selects paths through the WWW to relevant content. A number of current approaches to focused crawling are discussed below.

3.6.2.1 Link Prioritisation

The link prioritisation approach to focused crawling attempts to order the URLs to download so that the most desirable or relevant pages are downloaded first. Fish-Search [De Bra & Post 94b] and Shark-Search [Hersovicia et al. 98] were two of the earliest systems to attempt to prioritise the URL queue for a focused web crawl to improve the efficiency of the crawl.

The Fish-Search algorithm [De Bra & Post 94a] conducts a type of focused web crawl for each search query. It takes one or more starting URLs and the user's query as input. The queue of URLs to be downloaded takes the form of a prioritised list. The first URL in the list is taken and downloaded in a recursive fashion. The text of each page is analysed and assigned a rating in relation to the user's query. This rating dictates whether a page is deemed relevant and whether its links are scheduled for download. This was an extremely innovative technique and was much more efficient method of crawling for a specific topic than a

traditional breadth-first crawl. However, as a crawl was conducted for each search performed, it placed a very heavy load on web servers [Micarelli & Gasparetti 07].

Shark-Search [Hersovici et al. 98] is a more aggressive variant of the Fish-Search algorithm. In Fish-Search, regions of the WWW where relevant pages are not quickly discovered are discontinued. Shark-Search overcomes some of the limitations of this approach by measuring page relevance more precisely than the binary relevance function in Fish-Search. Shark-Search also makes finer estimates of the relevance of neighbouring pages and prioritises relevant pages or pages that are most likely to lead to relevant pages. In Shark-Search the potential relevance score of a link is influenced by its anchor text, the text surrounding the hyperlinks. Relevance is also influenced by an inherited score from incoming links.

WTMS [Mukherjea 00] is a system for gathering and analysing collections of web pages on related topics. WTMS contains a focused crawling component. This focused crawler assigns each page a representative document vector (RDV) based on the frequently occurring keywords in their URLs. A vector space model is used to compare discovered pages to a set of user-defined seed pages. URLs from pages which attain a similarity score above a specific threshold are queued for download. WTMS exploits the theory of topic locality through the inclusion of a "nearness" coefficient.

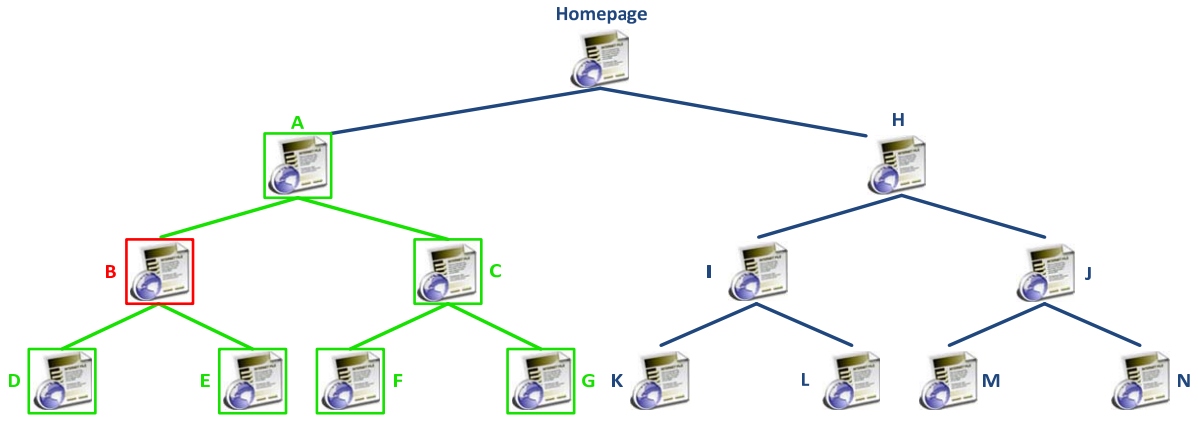


Figure 3-3 WTMS Nearness Graph

WTMS attempts to strike a balance in its estimation of nearness. If strict criteria for determining the nearness between two pages are used, the number of pages downloaded will be lower and some relevant pages may be missed. Conversely, lenient criteria will result in more pages retrieved but at the cost of increasing the number of downloads and the possible dilution of relevant content. The criteria used by WTMS are those shown in the above graph,

Figure 3-3. Only pages within the parent and sibling sub-trees of a site graph are downloaded as these are considered near. In the example displayed, when the focused crawler resides at page B, then only pages within the directories A, C, D, E, F and G are considered near and scheduled for download. This was believed to be the optimum setting to capture most of the relevant documents within a site without having to download every page on the site.

Other approaches to prioritising the crawl order of unvisited URLs have been researched and comparatively evaluated [Cho et al. 98]. Breadth-first is a common algorithm used in the IR community. If the WWW is represented as a graph of nodes, symbolising pages, and edges, symbolising hyperlinks then the order in which pages are visited is dictated by page depth. Pages at the same depth as the current page are visited before delving any deeper into a website. The Backlink count metric ranks each page in importance based upon the number of other crawled pages which link to it. This is a simple web-based implementation of bibliometrics. PageRank is a more sophisticated bibliometric method which uses an iterative algorithm to assign importance to each page based not only on the number of backlinks it receives, but also taking into account the relative importance of each page which links to it. PageRank has been examined in more detail in section 3.4.1.

Comparative evaluation of these approaches demonstrates that PageRank outperforms both Backlink and the random ordering if the goal is to crawl the most popular pages. Backlink tends to be more influenced by the seed set of URLs, becoming very locally focused and not discovering sites outside of the clusters where the seed URLs are located. However, if the goal is the discovery of pages relevant to a specific topic, the evaluation shows that Breadth-first performs most successfully [Cho et al. 98] [Micarelli & Gasparetti 07].

3.6.2.2 Taxonomic Crawling

A much more comprehensive approach to focused crawling on the WWW was proposed in [Chakrabarti et al. 99]. This focused crawling system has three constituent elements: a classifier, a distiller and a crawler. The classifier assesses the text of all the pages discovered for relevance to the purpose of the crawl. The distiller attempts to identify hubs through an implementation of the HITS algorithm [Kleinberg 99b]. Hubs are pages which provide links to many authoritative sources on the topic in question. The crawler conducts the content download, link extraction and manages URL prioritisation. It is governed by both the classifier and the distiller.

A canonical classification tree is generated from a topic taxonomy. The user can select the most applicable nodes, or leaves, from this tree in relation to the subject of the crawl. Once content is discovered and analysed, the classifier places it within the best matching leaf of the tree. If this leaf has been tagged by the user then the content is deemed relevant and its links are scheduled for download in the URI queue.

This focused crawling approach has since been further refined with the addition of a fourth component, the apprentice [Chakrabarti et al. 02]. The apprentice acts as a second classifier which prioritises the URLs in the queue, or crawl frontier. The idea behind the apprentice is to try and emulate human behaviour when browsing the web. On the WWW "every click on a link is a leap of faith" [Leiberman et al. 01], however humans are much more adept at using a variety of visual and textual clues to estimate the worth of the target page. The apprentice extracts features relating to a link from the Document Object Model (DOM) of the page from which the link was sourced. It then takes these features into account when prioritising the URLs in the crawl queue.

3.6.2.3 Context Graphs

Contextual graphs provide an alternate approach to the prioritisation of pages along a crawl path [Diligenti et al. 00]. An algorithm is used to build a representative model of the pages that occur within a defined link distance of a set of target pages on a specific topic. Using existing search engines, the WWW is *backcrawled* to build the context model. In other words, searches are performed to find pages which link to the seed set of target pages. A context graph is created which consists of all the discovered pages and their links in relation to the target pages. A relational value is calculated for each page. This value is defined as the minimum number of links it is necessary to traverse, in order to reach one of the original target pages.

This context graph is used during a crawl by a set of Naïve Bayes classifiers which can then make an attempt to estimate the link distance from any generic page to a relevant page. This context awareness allows the crawler to continue on a crawl path even if the reward for following a link is not immediate, but several links away. However, links which are expected to lead more quickly to relevant pages are favoured by the crawler. A major limitation of this

approach is the reliance upon the provision of sufficient backlinks to the seed set of target pages by an existing search engine.

3.6.2.4 Reinforcement Learning

Reinforcement learning is the name given to a machine learning framework which promotes optimal sequential decision making using rewards and punishments [Kaelbling et al. 96]. In reinforcement learning approaches to web crawling, the actions that generate a benefit during a web crawl are mapped. As a crawl progresses, more and more actions and their subsequent outcomes are added to the map. The map thus becomes influential in predicting which actions the system should undertake to achieve the greatest benefit.

In a focused crawling scenario the system identifies patterns of text in, and in close proximity to, hyperlinks. The text analysed includes the headers and titles of the page, in addition to the anchor text and text in close proximity to the anchor. The relevancy of the content linked to by previously encountered hyperlinks containing similar patterns can be used to determine which links are most likely to lead to relevant pages. An evaluation of this approach [Rennie & McCallum 99] produced impressive results. The evaluation used the crawler to harvest research papers from four Computer Science department web sites, in addition to pages relating to the officers and directors of 26 companies, from their websites. For each crawl training sets must be generated in advance and both transition and reward functions defined. These functions are used to calculate a value for each hyperlink in the collection which quantifies the benefit of following that hyperlink.

3.6.2.5 Intelligent Crawling

An "Intelligent Crawling" framework has been proposed [Aggarwal et al. 01], in which a classifier is trained as the crawl progresses. The aim of this approach is to statistically learn the WWW's link structure while performing a search. For each crawl or "resource discovery", a set of configurable "predicates" are defined, and subsequently used to estimate the probability that an unseen page will satisfy the information need. The predicates can be simple keywords, topical searches using hypertext classifiers, page-to-page similarity measures, topical linkage queries or any combination of these factors.

In intelligent crawling no linkage structure of the WWW, such as topical locality, is assumed. Instead, the crawler is tasked with gradually "learning" the linkage structure of the portion of

the WWW it is traversing. A key assumption of this approach is that different features of the WWW will be more useful in assessing the relevance of any given page depending on the predicates defined to guide the crawl. For some predicates the unseen page's URL may be most valuable in predicting relevance, while for others it may be most worthwhile to examine the content of the linking page. The crawler is expected to discern such feature values over the course of a crawl.

The crawler is not seeded [Aggarwal et al. 01] like other focused crawling techniques, rather it begins at general points on the web and gradually begins to auto-focus as it encounters content which satisfies the pre-defined predicates. The crawler initially behaves like a general purpose crawler, following all the links it encounters, but gradually it becomes more adept at selecting links which are more likely to lead to pages which are relevant to the user-specified predicate. However, the selection and refinement of the predicates which drive the entire crawl could be quite a complex task [Micarelli & Gasparetti 07]. It would be difficult for a non-technical educator to describe an entire educational subject area using such predicates.

3.6.2.6 Genetic Algorithm-Based Crawling

Genetic algorithms are inspired by the principles of Evolution and Heredity. In evolutionary biology, populations of species evolve with particular chromosomes and genetic structures. The species which are most suited to their environment survive longer and thus have an improved chance of reproducing and spreading their chromosomes. Species which are ill-suited to their environment tend to die out. This process is known as "natural selection". Similarly, when applied to Information Retrieval, genetic algorithms employ a number of potential solutions which "evolve" through a set of operators such as inheritance, random mutation, crossover etc. The solutions which perform most effectively are retained while the ineffective methods are discarded.

InfoSpiders [Menczer & Belew 00], utilises a collection of autonomous goal-driven crawlers without global control or state in the style of genetic algorithms. This system is also known as ARACHNID which stands for Adaptive Retrieval Agents Choosing Heuristic Neighbourhoods for Information Discovery. These evolving crawlers or "agents" traverse the WWW in response to user-defined queries, attempting to mimic the behaviour of a human surfing the web. Each agent autonomously assesses the relevance of any given page to the query and calculates the most desirable crawl path to follow.

The user initially provides a list of keywords and a list of seed URLs. The collection of crawlers is initialised by pre-fetching each of these seed pages. Each agent is randomly allotted one of the seed URLs as its start point and given a random behaviour and an initial supply of "energy". The energy that a particular agent has dictates its survival. Agents are awarded energy if a crawled page appears to be relevant and are deducted energy for all network loads incurred. Thus, if an agent does not regularly reach pages which are deemed to be relevant, it will run out of energy and "die".

Each agent analyses the content of its seed page and uses this analysis to determine the relevance of the sites linked to from the seed. Based on these estimates the agent creates a crawl order. An agent can modify its crawling behaviour based on previous results, by learning which links are most likely to lead to relevant pages. Agents can be selected for reproduction. When two agents reach a page at the same time, they can be combined to produce offspring. Random mutation can also occur, which helps prevent agents from converging on sub-optimal solutions.

A key element of this approach is the ability for the user to provide optional relevance feedback. The user can assess the relevance of the pages visited by the crawler. These relevance assessments can be made during the course of a crawl and alter the subsequent behaviour of an agent. The process is described as being "akin to the replenishment of environmental resources; the user interacts with the environment to bias the search process" [Menczer & Belew 00].

Another crawler based upon a genetic algorithm approach is the Itsy Bitsy Spider [Chen et al. 98]. This follows a similar process to InfoSpiders with some functional variations. The termination of an agent occurs when two consecutive generations do not produce improved relevancy with regard to retrieved content. There is no notion of rewards and penalties for choosing relevant content. There are similar evolutionary operators which perform crossover reproduction and mutation to evolve the agents. Mutation is achieved by utilising Yahoo's web directory to suggest possible promising new seed sites to individual agents. A Jaccard fitness function is used to assess the relevance of discovered pages. Each page is represented

as a weighted vector of keywords and compared to the corresponding vectors for the seed set to determine the probability of it being relevant.

However, despite its unique approach to focused crawling, genetic algorithms have not fared well during evaluation. An extensive study [Menczer et al. 01] which compared three focused crawling techniques, Best-first, PageRank and InfoSpiders, found the Best-first approach to show the highest harvest rate. During an evaluation of the Itsy Bitsy Spider approach [Chen et al. 98] the genetic algorithm approach again fails to improve upon the performance of a best-first search system. The recall values of the genetic algorithm approach are significantly better than the best-first system, but precision, which is most important in terms of the WWW, is not significantly improved.

3.6.2.7 Social or Ant-Based Crawling

Research into social insect collective behaviour has inspired a different approach to focused crawling [Gasparetti & Micarelli 03] [Gasparetti & Micarelli 04]. This model for focused crawling is based on how insects, in particular ants, help each other to find the way by marking trails with a hormone that can be detected and followed. In a similar manner a collection of crawling agents traverse the web and mark paths to relevant pages for other agents to follow. The assumption is that these agents upon exploiting this trail can make further explorations and discover more distant, relevant material than would have been otherwise possible.

This approach also addresses the concept of context paths [Mizuuchi & Tajima 99]. Individual web pages are often not self contained (although as the navigation methods employed by users evolve, this is now less frequently true; see section 2.3.4). The author of a page sometimes assumes that the browser has a certain degree of knowledge in a subject area or has browsed through other sites to get to this page. Sometimes in order to satisfy an individual's information need, it is necessary to provide information from a variety of sources. These pages, when connected, can be presented as a single unit of information, or a context path.

Each path that is followed by a crawling agent is marked by a trail. Each trail is rated, or assigned a "pheromone intensity", dependant on the number of relevant resources found on a path, and the distance the crawler must travel to reach these resources. Each crawl is cyclical

and a drawback to this approach is that the crawler must begin each cycle at the same collection of seed pages. This can lead to a large amount of duplicate crawling of pages and a subsequent waste of computational resources.

3.6.3 Analysis

Focused crawling has been shown to be a powerful means for the discovery of topic-specific resources on the WWW. Such crawlers can be driven by a variety of different means including keyword descriptions or exemplar documents. Some crawlers also attempt to learn what constitutes a relevant document as a crawl progresses, through human-aided relevance feedback or machine learning. These crawlers then explore the Web, guided by relevance or popularity assessment mechanisms. Content is filtered at the point of discovery rather than post-indexing which improves its efficiency with regard to computational resources. Focused crawlers provide a means of exploring the WWW in a controlled, subject specific and yet dynamic fashion.

The numerous approaches to focused crawling described in the above section all use distinct techniques to discover relevant content. There are various aspects and limitations of each approach that should be kept in mind during the design of a focused crawling service for TEL.

When link prioritisation is applied to subject-specific crawling, evaluations have shown breadth-first to be the most successful link-ordering algorithm. Taxonomic crawling could be used to allow a subject matter expert to define the crawl limits. However, this must be implemented in such a manner that it can be conducted by a non-technical user. The context graph approach is limited by it reliance upon existing search tools to backcrawl the web. In reinforcement learning, the definition of sets of transition and reward functions, which are necessary in this approach, could be difficult for a non-technical user. In intelligent crawling, the crawler starts by conducting a general purpose crawl and gradually focuses by following links which are more likely to lead to relevant content. These crawls will, by nature, be longer and more demanding on computational resources. The selection and refinement of the predicates which drive the entire crawl could also be quite complex. It would be difficult for a non-technical educator to describe an entire educational subject area using such predicates. When evaluated for focused crawling, genetic algorithm techniques fail to improve upon

standard, best-first techniques. Social or ant-based crawling techniques tend to result in large amounts of duplicate crawling and unnecessary resource consumption.

3.6.4 Summary

This section detailed the current trends and approaches to the implementation of focused or topic-specific crawling on the WWW. The section examined the underlying theory of topic locality and how it relates to focused crawling. It then went on to analyse in turn the various approaches to performing the focussing, or relevance assessment, aspect of a web crawl. In the following section various open source implementations of web crawlers will be examined.

3.7 Open Source Web Crawlers

It is a goal of this research to utilise open source software, where possible, in an effort to avoid "recreating the wheel". There exist a number of web crawling systems which are open source and available for integration into individual research projects. All boast diverse and varying feature sets that satisfy the requirements of specific online communities. An objective of this research is to identify IR techniques and technologies which can be successfully applied within TEL. Conducting an analysis of the web crawlers most applicable to this research which are in active use was an essential precursor to the design of a focused content retrieval mechanism for this research. These web crawling tools are discussed in the section below.

3.7.1 i-Via Nalanda

The Nalanda focused web crawler [Nalanda] is an open source focused crawler developed by IIT Bombay and the U.S. Institute of Museum and Library Services for the iVia Virtual Library System project. It is based on the research conducted in [Chakrabarti et al. 99] [Chakrabarti et al. 02]. The crawler supports focused crawling on a predefined subject domain and employs two methods of classification to improve the accuracy of content topic matching.

Nalanda exploits the fact that resources focused around a common topic often cite one another, as discussed in sections 3.4.1 and 3.6.1. Highly inter-linked resources are examined, evaluated and rated for their capacities as authoritative information sources. These content sources are split into authoritative resources and hubs. The second method of classification

that Nalanda employs is a keyword and vocabulary comparison between candidate resources and resources that have already been accepted into the collection. This classification occurs after the linkage analysis has been performed.

The crawler has been improved with the addition of an 'apprentice' component. The apprentice is a learning algorithm which intelligently recognises clues in hyperlinks to try and decide upon the most promising links to follow for the purpose of each crawl.

3.7.2 Combine Harvesting Robot

Combine [Combine] is an open source system for crawling, harvesting and indexing internet resources. It was initially developed as a general purpose crawler as part of the Development of a European Service for Information on Research and Education (DESIRE) project [Desire]. It was later modified for use as a focused crawler by the EU project ALVIS – Superpeer Semantic Search Engine [Ardö 05].

The general purpose web crawler has been combined with an automated subject classifier created by the KnowLib research group [KnowLib] to generate topic-specific databases of crawled information. The crawl focus is provided by the use of an ontology that is used for topic definition and term matching. When a document has been deemed relevant, further processing, such as character set normalization, language identification and simple text segmentation, is performed in preparation for processing of the data.

All crawled information is stored locally in a relational database. By using this central database for synchronisation, it is possible to run several crawlers in parallel, all crawling a single subject domain. Combine can be used for metadata extraction. It can handle multiple document types including plain text, HTML, PDF, PostScript, MsWord, LaTeX and images. A SQL database is used for central data storage and administration.

Before a crawl can be conducted the user must create what is termed a topic definition. The topic definition breaks each subject down into hierarchical subject classes. A file is then created by the user which contains keywords, phrases or boolean expressions which are related to the subject area. Each of these items must be associated with a subject class and have a term weighting applied. Term weights can be positive or negative. A list of seed URLs must also be collated and provided to the crawler as start points for the crawl.

3.7.3 Heritrix

Heritrix [Mohr et al. 2004] is the Internet Archive's [Internet Archive] open source, extensible, web-scale, archival quality web crawler and is available under the GNU Lesser General Public License [LGPL]. The crawler was initially developed in 2003 and has since become a stable platform with a large community of users and developers that perform regular bug fixes and provide assistance through a lively mailing list. The crawler is implemented in Java and is designed to be able to perform broad crawling, focused crawling, continuous crawling and experimental crawling. For more detail on the implementation of these crawling methodologies see section 5.2.3.1.

The crawler follows a standard crawling methodology using a pluggable architecture. A URI is chosen from among those scheduled and the content is fetched. The content is then either archived or analysed depending upon the purpose of the crawl. Any URIs in the content are extracted from the retrieved content and added to the queue. The URI is then marked as complete and the process is repeated recursively. The Heritrix architecture is divided into three main components. The scope, which defines which URIs should be crawled. The frontier, which monitors the queue of URIs and schedules those to be downloaded. The final component is the processor chain which is composed of pluggable components which each perform an operation upon the content. These components can perform either pre or post-processing tasks or content analysis tasks.

An example of a system that used Heritrix to perform focused crawling tasks was Metacombine [Metacombine]. Metacombine was a Mellon Foundation-funded project hosted at Emory University to research methods to more meaningfully combine digital library resources and services. This project worked to combine metadata technologies such as OAI-PMH with web crawling techniques in an effort to improve scholarly communication. Heritrix was used to generate collections of web resources relating to the American south for use in the project.

3.7.4 Summary

In the previous sections, IR on the WWW as a discipline was introduced. The emergence of specific techniques and methods in the area of web crawling were detailed. Focused crawling was then comprehensively examined as a means of traversing the web in a topic-specific

fashion. In this section five publicly available web crawling solutions were analysed and detailed as a precursor to the creation of a focused content retrieval mechanism for this research. The next section of this chapter will address post-crawl content discoverability issues, such as indexing and searching.

3.8 Indexing and Searching Content Sourced from the WWW

The representation of the content contained within documents and, in the case of Information Retrieval on the WWW, web pages, is referred to as indexing. Indexing is a process which translates natural language content into a machine usable form. This involves methods of deducing which lexical terms best describe the content. The index which is generated by this process is then used in combination with a user query, by a search engine or other such service, to locate and rank relevant information [Salton & McGill 84] [van Rijsbergen 79].

3.8.1 Indexing

The explosive growth of the WWW, as discussed in section 3.5, has made the user-driven discoverability of content, and consequently the indexing of content, increasingly important. The indexing mechanism is a critical component of any web-based IR system. It must provide a formalised, simplified and machine usable representation of the natural language content contained within each web page [Salton 89]. The following sections will detail the most common approaches to indexing in the IR field and subsequently, how these approaches have been adapted to deal with the issues and challenges of representing web content. The typical processing conducted on pages will be explained followed by the some of the widely applied indexing, or document modelling, techniques.

In traditional IR approaches to content indexing, each document is treated as an unstructured, unordered bag of words. Indexing is based upon the assumption that the occurrence of terms within a document can be used to determine the subject matter of the content. Pre-processing steps are conducted to simplify this term-based analysis. Pre-processing consists of a series of steps aimed at removing all information unrelated to the semantics of the content. Once this is complete, terms which are deemed meaningful are extracted from the document and weights are calculated for each. These weights are used to signify the importance of the term as an indicator of the documents subject matter. This allows the IR system to discriminate between documents with respect to terms in a user query, rank the documents according to relevance and present the most relevant documents to the user.

3.8.1.1 Term Normalisation

In traditional IR approaches, term normalisation occurs as one of the document pre-processing tasks. Term normalisation refers to the conversion of a page from a structured written work, into an un-structured stream of text. This means the removal of all text cases and punctuation.

3.8.1.2 Stopword Removal

Not all terms aid discrimination between pages when searching an index. Some terms may be descriptive of the subject matter of an individual page, but occur so regularly in a collection of documents that they are poor discriminators between pages. Words are not evenly distributed across languages, be they in written or spoken form. Few words appear very frequently, while many words appear very infrequently. This distribution of words is known as Zipf's distribution [Zipf 35] [Zipf 49].

According to Zipf's law, when attempting to differentiate documents in a corpus, high frequency words hold little value. They describe too many individual pages in the corpus and add no unique descriptive value. Zipf's law also states that words which occur extremely infrequently in a corpus may also be of little value. They may be spelling mistakes or uncommon proper nouns. These words are too rare to be of value. The most valuable terms for discriminating between documents within a corpus are those which occur with medium frequency. Frequently occurring terms, particularly connectives, articles and prepositions are commonly removed from an IR index. These terms are known as stopwords. Stopwords can be defined as any terms which add no, or limited, value to an index. This process reduces the number of common terms between pages in a collection.

3.8.1.3 Stemming

The process of stemming conflates morphologically similar terms to their morphological root, so that the index can recognise variations of the word when conducting a search [Lovins 68] [Porter 80] [Sparck Jones & Willet 97]. Words can occur in various forms and all variations need to be recognised as referring to a common concept. This process can aid the number of relevant pages returned for a search. A typical stemmer consists of a collection of rules and dictionaries [Krovetz 93]. These rules are specific to each language used and can be extremely complex. Care should be taken when defining such rules as there can be problems with the stemming process. Terms can be produced which are not actual words and the

process can be difficult for a user to understand. For example, when using the Porter Stemming Algorithm [Porter 80], computer is conflated to comput and ponies to poni.

It is important to have a balanced approach to stemming so that the majority of the benefits of the process are reaped without unduly hampering retrieval performance. The stemming process also has the desirable side effect of reducing the index size. The process of stemming increases the number of common terms between resources in a collection. This improves the recall of IR systems. Stop-word removal on the other hand, reduces the number of common terms between resources in a collection. This aids discrimination between resources and improves precision.

3.8.1.4 Term Weighting

Early IR systems merely recorded the presence or absence of a term. This is known as binary retrieval. More advanced IR systems weight terms according to their relative estimated importance. This importance measure can be derived from a term's importance within a corpus of content or a term's importance within an individual page. These measures can be influenced by each other. If a term occurs frequently within a corpus, its importance within an individual page is less significant. For instance, the terms "Dublin" and "Traffic" occurring frequently in a page are of little importance if the corpus being indexed is a collection of Dublin Transport Office web pages. Term frequency (*tf*) is used to measure the importance of a term in an individual document. *tf* is defined as:

$$tf_{dt} = \left(\frac{num_t}{total_d}\right)$$

Where $total_d$ is the total number of terms in a document d and num_t is the total number of times that a term t occurs in d. *tf* returns high values for frequently occurring terms. *tf* is calculated for all index terms on all documents in the corpus. A Term-Document Frequency Matrix can then be constructed for the collection. Inverse document frequency (*idf*) is used to measure the importance of a term within a corpus of content [Sparck Jones 72]. *idf* is defined as:

$$idf_t = \log\left(\frac{n}{n_t}\right)$$

Where t is a term which occurs in a collection of documents. n is the number of documents in the collection and n_t is the number of documents in the collection in which t occurs. *idf*

returns high values for infrequently occurring terms. Term frequency and inverse document frequency are often combined in IR to provide a more accurate overall term weight (*tf-idf*) [Salton & Yang 73] [Salton & Buckley 88].

$$weight_{dt} = idf_t * tf_{dt}$$

A high *tf-idf* value is achieved if a term has a high frequency within the document in question, but a low frequency in the corpus as a whole. A proposal has been made to adapt the *tf-idf* method and apply it to the anchor text of hyperlinks on the WWW [Hawking et al. 04]. Other weighting schemes have been developed with particular IR models in mind. For instance, a method based on genetic programming has been defined [Cummins & O'Riordan 06] which can automatically determine term weighting schemes for the Vector Space Model (VSM). The VSM will be discussed in more detail in section 3.8.2.2. This is based on a set group of queries and a user-selected collection of relevant documents for the queries. Weighting schemes are then evolved which achieve a high level of retrieval precision on the corpus. Schemes are evolved in document specific (local) and corpus specific (global) domains and combined to produce the best results.

3.8.1.5 Analysis

Term frequency and term weighting techniques will be particularly important when creating caches of subject specific content for use in educational scenarios. As a collection of content will generally be on a single subject, some terms which are very valuable during a web crawl, will be less valuable during indexing. For instance, if collecting content related to "The Poetry of W.B. Yeats", the term "Yeats" will be extremely useful when searching for relevant content on the open WWW during a crawl. However, when later searching across a collection of content consisting solely of pages related to Yeats' poetry, the value of the term "Yeats" may be greatly reduced. More detail on each of these page pre-processing steps, part-of-speech tagging and term weighting, using practical examples, can be found in Appendix B.

3.8.1.6 Summary

Indexing is the process of representing content so that an IR system can discriminate between resources with respect to a user's information need. The IR system uses the index to make a relevance assessment and present the most relevant resources to the user. This section has provided the reader with an introduction to the most successful approaches to indexing in the IR field. Content pre-processing is examined and the processes of normalisation, stopword

removal and stemming are explained. Finally the section concludes with an detailed examination of approaches to index term weighting. In the next section contrasting methods of calculating query-document similarity are presented. These approaches to matching index terms to a query are called IR models. The models examined are the Boolean model, Vector Space model, Probabilistic model and Language Model.

3.8.2 Information Retrieval Models

IR models form a description of the computational process of retrieval. This includes the process by which an information need is initially articulated and possibly refined. The model also defines the method of selecting a document from among a collection for retrieval. The IR models examined here are the most popularly applied models. Each is based upon user generated queries which consist of one or more keywords. The semantics of the resources in a collection and the user information need can both be expressed through combinations of such terms. The relevance of a resource can then be estimated using similarity functions which compare the information need and the available resources.

3.8.2.1 Boolean Model

The Boolean Model approach to IR is based upon Boolean Logic, Boolean Algebra and Set Theory. It is one of the oldest IR models. In this approach, pages are represented as a set of keywords which have been automatically or manually extracted from all the pages in a collection [Salton & McGill 83] [van Rijsbergen 79]. Each page in the collection is represented as a vector:

$$\overrightarrow{d_p} = \{(t_1, w_{1p}), (t_2, w_{2p}), \ldots\ldots, (t_n, w_{np})\}$$

A binary value is assigned as the weight w, of a term t, based upon its occurrence or non-occurrence in a document d. The weight is set to 1 if the term occurs in the document in question and the 0 if it does not occur. In the Boolean Model each query is represented as a Boolean expression of terms and their connectors (AND, OR etc.).

3.8.2.2 Vector Space Model

A vector defines a position in space. In the Vector Space Model (VSM) approach to IR, each document is represented as an n-dimensional vector. There is one dimension for each index term in the collection. Calculated term weights or Boolean values can be used. This approach was first implemented in the SMART retrieval system [Salton 71]. Each document in the

collection undergoes all the pre-processing steps already described above. Once the pre-processing steps are complete, a Term-Document Frequency Matrix is generated for the collection.

Generally the vector for each document is sparse in nature, i.e. it contains a very small subset of the full list of terms. Every term $t_i \in d_j$ has a weight w_{ij}, such that: $w_{ij} > 0$ *if* $t_i \in d_j$ and t_i does not belong to all documents in the corpus [Micarelli et al. 07]. These weights are calculated using some measure of relevance, such as *tf-idf*. Terms which occur in all documents in the collection are excluded from the vector generation. Each co-ordinate of the vector space corresponds to an index term in the corpus and its importance in terms of that document. Queries are also represented, using the same rules as documents, as an n-dimensional vector.

$$\vec{q} = \{w_{1q}, w_{2q}, \ldots, w_{nq}\}$$

Once a query has been represented as a vector in the same n-dimensional concept space as the document corpus, relevant documents can be retrieved using very simple similarity functions. Cosine Correlation is the most employed of these similarity functions.

$$sim(d_j, q) = \cos(\vec{d_j}, \vec{q}) = \frac{\vec{d_j} * \vec{q}}{|\vec{d_j}| \, |\vec{q}|}$$

This similarity function is equal to the cosine of the angle formed by the vectors $\vec{d_j}$ and $\vec{q}$.

3.8.2.3 Probabilistic Model

The Probabilistic Model attempts to estimate the probability that a user will find a given document relevant to their query. This captures the IR problem within a probabilistic framework [Robertson & Sparck Jones 76] [Sparck Jones et al. 00a] [Sparck Jones et al. 00b]. The Probabilistic Model employs binary weight vectors in a similar fashion to the Boolean Model. However the query-document similarity function in the Probabilistic Model is more complex.

By applying Bayes rule [Bayes 1763] and odds rather than probabilities [Micarelli et al. 07], retrieved pages are ranked by the odds that they are relevant. This is equivalent to the probability that the page is relevant to the query divided by the probability that the page is not relevant to the query. The probabilistic model is a recursive function and requires initial

estimated assumptions of relevancy as there are no retrieved documents upon which to base the probabilities. The function then recursively refines these initial estimations to obtain the final ranking of pages based upon their relevance probability. For a detailed derivation of the algorithms employed in the probabilistic model, the reader should refer to Appendix B.

3.8.2.4 Language Model

The Language Model estimates query-document similarity in a different manner to the other mainstream IR models. When searching for information, users commonly form queries consisting of terms which they expect to appear in relevant documents or web pages. The Language Model attempts to model this process. When searching across a collection of documents, an individual document is considered relevant if modelling that document is likely to produce the query in question. Unlike the Probabilistic Model which calculates the probability that a document is relevant to a given query, the Language Model approach builds a probabilistic language model for each document in the collection. Documents are then ranked based upon the probability of each model generating the given query [Ponte & Croft 98] as shown below:

$$P(Q|D) = P(q_1, q_2, \ldots, q_n|D) = \prod_{t=1}^{n} P(q_t|D)$$

The last term is obtained from the assumption of conditional independence of terms given the document's language model. Since a document or web page is typically larger than a keyword query, estimating a language model for a document is a simpler task than estimating a model of relevant documents based on a query. Thus, the Language Model approach avoids the problem encountered by the Probabilistic Model of generating initial estimated assumptions of relevancy.

The Language Model approach has similarities with traditional tf-idf methods. Term frequency is directly represented in tf-idf, and much recent work has defined the importance of document length normalisation [Karbasi & Boughanem 06]. The combination of document generation probability with collection classification acts in a similar fashion to idf. Terms which are rare in the document collection but common in individual documents will be more influential when estimating relevancy. Both models also share the assumption of term independence. However, in terms of performance, the Language Model significantly

outperforms pure tf-idf measures [Manning et al. 08], and has typically equalled the performance of the Vector Space Model [Dai et al. 05].

3.8.2.5 Web-based Indexing and Retrieval

When applied to the WWW, the traditional approaches to IR and indexing discussed above can be further refined to account for the structure of web pages [Agosti & Melucci 00] [Kobayashi & Takeda 00]. Using the tag structure of HTML, and similar XML-based content formats, it is possible for term weights to be altered based upon where the terms occur within the page. For instance, a term which occurs once in the title of the page is much more indicative of the subject matter than a term occurring multiple times in a paragraph further down the page.

This approach can be extended to extract distinct semantics from the page structure or to deduce term importance based on the physical characteristics of a page. Take for example scholarly research papers on the web which are often organised into sections such as *Title, Authors, Introduction, References* etc. Information regarding the nature of the terms and how they inform the subject matter of the entire page can be deduced from the section in which they appear [Cutler et al. 99] [Molinari et al. 03]. This takes into account the entire syntactic structure of a HTML page. Term weights can be calculated based upon the importance of the tags in which they appear in relation to the structure of the page. Traditional IR weighting methods, such as *tf-idf*, are used to calculate the importance of a term within a tag. These values are then aggregated according to the importance of each tag in the page. Example tag hierarchies are shown in figures 3-4 and 3-5 below.

Rank	Class Name	TAG / Parameter
1	Title	TITLE, META Keyword
2	Header 1	H1, Font Size=7
3	Header 2	H2, Font Size=6
4	Header 3	H3, Font Size=5
5	Linking	A HREF
6	Emphasised	EM, STRONG, B, I, U, STRIKE, S, BLINK, ALT
7	Lists	UL, OL, DL, MENU, DIR
8	Emphasised 2	BLOCKQUOTE, CITE, BIG, PRE, CENTER, TH, TT
9	Header 4	H4, CAPTION, CENTER, Font Size=4
10	Header 5	H5, Font Size=3
11	Header 6	H6, Font Size=2
12	Delimeters	P, TD, Text not included within any Tag, Font Size=1

Figure 3-4 Tag Hierarchy for Term Weight Calculation [Molinari et al. 03] [Marques Pereira et al. 05].

Class Name	HTML Tags
Anchor	A
H1-H2	H1, H2
H3-H6	H3, H4, H5, H6
Strong	STRONG, B, EM, I, U, DL, OL, UL
Title	TITLE
Plain Text	None of the Above

Figure 3-5 Tag Hierarchy for Term Weight Calculation [Cutler et al. 99].

The typographical appearance of the content within a web page can be used to discern term significance within the page. Character dimension and emphasis, such as underlining and font style, can be used to influence term weighting [Bordogna & Pasi 00]. This approach has been extended [Marques Pereira et al. 05] to add a contextual element to the term weighting process. The contextual aspect of the process allows for the consideration of the style of HTML tag distribution in the documents of a corpus.

Some approaches have combined traditional IR indexing and retrieval methods with HTML tag-influenced weighting systems and link-analysis algorithms like PageRank and HITS, discussed previously in section 3.4.1. One particular fusion approach [Yang 01] combined

web-specific weighting measures with a modified HITS algorithm. Index term weight was calculated based upon the HTML tag in which the term occurs. For instance, the frequency of a term is increased by a factor of ten if the term occurs in the HTML header. This approached is then combined, using the Similarity Merge method, with the link-analysis and ranking methods of HITS. However, the authors found that fusing this weighting method with HITS link-based retrieval was not beneficial. The best text-based retrieval system in the trial not only outperformed the best link-based system, but also outperformed all fusion approaches also.

3.8.2.6 Analysis

This section reviewed four IR models with distinct approaches to query-document similarity estimation. Web-based methods of indexing and retrieval were also examined. The Boolean Model approach is quite simplistic and very easily implemented. However it is also quite limited in scope. It will only retrieve exact matches, and does not estimate relevancy. Either the query terms that are submitted all occur within a page or it will not be presented as a possible result. All exact matches are presented in an unordered list. There is also no weighting of terms in relation to their importance in a corpus. All terms are deemed to be of equal importance, a regularly invalid assumption, as was already shown in previous sections.

The Vector Space Model can be used to perform more subtle measures of resource relevancy for a search query than the Boolean Model. Documents do not have to be exact matches to be retrieved and presented to the user. Term weighting is applied for each term in the search query and an n-dimensional vector is created for every resource in the collection, as well as for the query. Retrieved documents can also be ranked for relevancy, most commonly using cosine correlation.

In the probabilistic model, retrieved documents are ranked according to the probability that they are relevant to a given query. This approach assumes the division of documents into relevant and non-relevant sets. Binary weights are employed in a similar fashion to the Boolean model. The probabilistic model fails to take into account the frequency with which terms occur within a document or within a collection. The model also imposes some restrictive simplifying assumptions such as term and document independence.

The Language Model applies probabilistic approaches to IR but in a different manner to the Probabilistic Model. The Language Model is a generative model and attempts to model the manner by which users generate queries to retrieve content. Term frequency is accounted for in this approach and an initial model of relevancy does not need to be estimated, unlike the Probabilistic Model. Based on current evaluations, the performance of the Language Model appears to be generally equivalent to that of the Vector Space Model.

These traditional IR techniques can be further supplemented by exploiting the standardised structure of web documents. As described in the above section, this can be achieved by altering the term weighting process. Such a supplemental approach to content indexing and searching could prove very useful in the development of a tool to search collections of educational content sourced from the WWW. The vast majority of educational content encountered will be in a structured form such as HTML, XHTML, PHP or similar. This would enable a finer level of relevance assessment across the collection of content and thus improve the retrieval quality.

3.8.2.7 Summary

This section has provided the reader with a detailed overview of three contrasting computational models for IR. The most basic, and oldest, of the three models in discussed first, the Boolean model. This is an approach based on Boolean Logic, Boolean Algebra and Set Theory. The Vector Space model is then detailed. In this approach each content resource is represented as an n-dimensional vector. Relevance assessment is then conducted across the vector space. The section concludes by examining both the Probabilistic Model and the Language Model, which estimate relevancy through the calculation of probabilities. The next section examines various approaches to web-based search and the user interfaces types used by these tools.

3.8.3 Retrieval Interfaces

When a user executes a search on the WWW, they are driven by some form of information need [Schneiderman et al. 97]. However, the user's intent when searching is not always informational; it can also be navigational or transactional. Navigational searches occur when a user is aware a site exists and needs to find its URL. Transactional searches occur when a user needs to make a transaction, such as a goods purchase, and needs to find a site where

they can perform this transaction [Broder 02]. Web IR Interfaces have evolved around satisfying these distinct information needs.

A diverse variety of interface types have become available as the number of web IR tools competing for traffic has grown. The majority of mainstream search engines are designed to satisfy information needs and employ a simple keyword-based search interface. Google [Google] has always been noted for its minimalist design which makes its interface very intuitive to use. Many other search engines have similar interfaces including AltaVista [AltaVista], Ask.com [Ask], Cuil [Cuil] and Bing [Bing]. Yahoo! [Yahoo!] has similar search functionality but couples this with a directory structure of the WWW which can be browsed. The Yahoo! homepage is much more cluttered as it acts as a portal to other functionality such as email and instant messaging. However, it does now have a dedicated search homepage which follows a minimalist design similar to Google's. Most mainstream search tools offer "advanced search" functionality which allows users to define more explicit queries. Parameters such as desired content format, the number of results returned and the language of content returned can be specified.

More recently, social search platforms such as Eurekster [Eurekster] have emerged. Social search allows the user to leverage the accumulated knowledge of communities via custom search portals called swickis. A user can build and customise a swicki on any topic. This portal can then be shared and distributed to grow a community of interested users. The swicki becomes more valuable and useful the more it is used. Yahoo! Answers [Yahoo! Answers] is another form of social search. It allows users to ask questions of the user base, post answers and browse popular question threads.

Spock.com [Spock] is a "people search" tool. It allows the user to use a keyword search input box to enter a name, email, location or tag to find information about individuals. Numerous "meta search engines" have appeared which attempt to leverage the power of established, popular search tools by combining their result sets. Dogpile [Dogpile] combines the search result listings of Google, Yahoo!, MSN Live Search and Ask in one result set.

Another variation of WWW IR tool that has emerged are "vertical" or specialist search engines, where the interface is tailored toward a particular community. Yahoo! Kids [Yahoo!

Kids] is a portal for children including basic keyword search and "Ask Earl" a form of tailored social search. Cranky.com [Cranky] describe their service as the first "age relevant" search engine and is targeted at the over 50's. TrueLocal [TrueLocal] and YourLocal [YourLocal] are transactional search tools which provide an interface where a user can specify a service and a location and receive a list of service providers at that location.

Other WWW IR tools are attempting to provide interfaces with innovative results visualisation methods. KartOO [KartOO] combines meta search functionality with an interactive Mindmap display of its results list. Snap [Snap] combines a traditional keyword search interface and ranked results listing with dynamic page preview functionality when a result is hovered over with the cursor. Tafiti [Tafiti] is an experimental search interface from Microsoft, powered by Silverlight, which employs various novel techniques in its interface, such as allowing the saving of individual results from a ranked list for future use. Tafiti also allows the display of results lists as branches of a tree which can be grown, shrunk or rotated.

3.8.3.1 Analysis

Despite the numerous search techniques and types of user interface available, various studies [Deepak & Parameswaran 05] [Bharat & Chang 03] [Quesenbery 03] have found that users prefer simplistic user interfaces when searching for information on the WWW. Users favour simple keyword based approaches where they can type in a query immediately and where query refinement is easy and quick. This is beneficial as a search proceeds through iterations or as a user's information need evolves. Notably, this approach is deemed more desirable than the use of pre-query formulation features to aid the production of queries which are more likely to retrieve accurate results. It was also found [Deepak & Parameswaran 05] that the "advanced search" functionality of web retrieval tools such as search engines are rarely used.

3.8.4 Indexing and Retrieval Software

As mentioned previously, this research aims, where possible, to utilise open source software solutions to address some of the requirements raised by this state of the art analysis. There are numerous open source indexing and retrieval software platforms available for use with collections of content sourced from the WWW. Conducting an analysis of some of the more successful solutions in active use was an essential precursor to identifying methods which could be applied within TEL, in the design of a content retrieval mechanism. These open source indexing and retrieval tools are discussed in the section below.

3.8.4.1 ht://Dig

The ht://Dig [ht://Dig] system provides open source indexing and search functionality available under the GNU General Public Licence [GPL]. The system was developed at San Diego State University as a means of indexing and searching the content on various web servers on the college network. ht://Dig uses fuzzy word indexing algorithms to index term occurrence in documents based upon different criteria such as synonym rules and word ending rules. The system provides a Common Gateway Interface (CGI) program for conducting searches. This can be invoked via a HTML form in a web page. There is also a command line option. Searches are conducted using keywords which are matched to the terms in the index. This system is specifically designed to cover the search requirements of a single company, campus or web site. It does not support web-scale indexing and searching.

3.8.4.2 Lucene

Lucene [Lucene] is a full-text indexing and search library available under the Apache Software Licence [ASL]. The flagship Lucene product is coded in Java, however Lucene has also been ported to various other programming languages including Delphi, Perl, C#, C++, Python, Ruby and PHP. Lucene is a standalone library which provides functionality for indexing and search. However it does not support parsing for the individual document formats that may be encountered in a selection of content sourced from the WWW. A separate parser needs to be provided for each document format, such as HTML, PDF, PHP etc. Once the text within a document has been parsed, Lucene provides several text analysis tools to aid the indexing process. These analysers facilitate pre-processing steps such as text normalisation, stopword removal and stemming. Lucene creates indexes, which are split into segments, and support *tf-idf* style term weighting. Query-Document similarity in Lucene is conducted using a combination of the Vector Space Model and the Boolean Model. The Boolean Model is used to initially filter the documents that are relevant based upon the use of boolean logic in the query analysis.

3.8.4.3 Lemur

Lemur is a toolkit for IR which provides indexing functionality based on the Language Model approach, although it can also be adapted to support other models such as VSM. Lemur is designed for a broad range of IR applications including ad hoc and distributed retrieval, cross-language IR, content filtering, and categorisation. The toolkit also provides basic search functionality using approaches such as Okapi [Karamuftuoglu et al. 02] and KL Divergence

[Lafferty & Zhai 01]. Lemur is open source and the project actively encourage users to modify the toolkit in support of their own research.

Lemur currently supports a variety of indexing functionality including: English, Chinese and Arabic language text; word stemming (Porter and Krovetz); stopword omission; acronym recognition; part-of-speech and named entity recognition. The Lemur toolkit provides a stand-alone GUI which provides keyword-based search. Lemur is implemented in C++, while the search interface is implemented in Java/Swing. It is compatible with UNIX, Linux and Windows.

3.8.4.4 Xapian

Xapian [Xapian] is an open source indexing and search library, released under the GNU General Public Licence [GPL]. It is implemented in C++, but provides bindings which enable its use from systems written in Perl, Python, PHP, Java, Tcl, C# and Ruby. Xapian is a highly adaptable toolkit which allows developers to easily add advanced indexing and search facilities to their own applications. Query-Document similarity is based upon the Probabilistic Model using an implementation of the BM25 or Okapi Probabilistic weighting scheme [Sparck Jones et al. 00a] [Sparck Jones et al. 00b]. Xapian also allows the use of a Boolean weighting scheme or the specification of customised weighting schemes. The search interface supports a rich set of Boolean query operators. Omega [Omega] is an open source search interface which can be used in combination with Xapian.

3.8.4.5 WebGlimpse and Glimpse

WebGlimpse [WebGlimpse] is indexing and search software for WWW content which is free to use within the education, government and not-for-profit domains and available via a licence for commercial sites. It is based upon Harvest [Harvest] a distributed search engine framework originally developed in 1995 at the University of Arizona. The software has several independent components including a web crawler, implemented in Perl, and Glimpse, which is the core indexing and search functionality, implemented in C. Glimpse, which stands for GLobal IMPlicit Search, generates indexes which are then available through a simple CGI interface. The number of terms used in the generation of a Glimpse index is configurable to control index size. The WebGlimpse search interface supports Boolean termed search and pattern matching for relevance.

3.8.5 Summary

In the previous sections post-crawl content discoverability was analysed and the related challenges outlined. Indexing and retrieval were introduced as component disciplines of IR. The emergence of specific techniques and methods in the field were detailed. Indexing approaches and Query-Document similarity models were then comprehensively examined as a means of discovering relevant content from a document collection. Five publicly available indexing and retrieval software solutions were analysed and detailed as a precursor to the creation of a content retrieval mechanism for this research.

3.9 Information Retrieval Tool-Chains

There are several existing Information Retrieval tool-chains available which combine web crawling, content indexing and query-document similarity matching functionality. One of the key goals of this research is to develop such a tool chain, specifically for the domain of education. An essential precursor to such development was the examination of these existing solutions to determine the range of functionality that each provide. As with the open source web crawling and content indexing solutions available, all boast diverse and varying feature sets that satisfy the requirements of specific online communities. The following section presents an overview of the most relevant tool chains to this research.

3.9.1 Nutch

Nutch [Nutch] is an open source search engine which provides both web crawling and web-based search functionality. Nutch is divided into two distinct components: the web crawler and the searcher. The web crawler fetches web pages and creates an index while the searcher identifies content relevant to a user's search query.

The crawler component of Nutch consists of three main components: a fetcher for content harvesting and link extraction; a custom database that stores URLs and the content of retrieved pages; and an indexer, which processes each page and builds a keyword-based index. Nutch is implemented in Java and is designed to support three distinct scales of crawling: local filesystem crawling; intranet crawling; and open web crawling.

A web database, or WebDB, mirrors the structure and properties of the portion of the WWW being crawled. Nutch defines this portion of the WWW as a web graph, where the nodes are

web pages and the edges are hyperlinks. The WebDB is used only by the Nutch crawler and does not play any role during searching. Nutch groups crawls into segments. Each segment consists of the collection of pages fetched by the crawler in a single crawl instance. Each segment has a fetchlist, which is a list of URLs for the crawler to fetch. This is generated from the WebDB. The fetcher step then requests web pages, parses them, and extracts links from them. Both the content of pages and URLs are stored in the database. Numerous pieces of information about each page are stored, such as: the number of hyperlinks in the page; fetch information; and the page's score, which can be calculated using selected link analysis algorithms. Nutch observes the Robots Exclusion Protocol when crawling the open web.

The indexing and search functionality of Nutch is based upon Lucene. The content parsing and indexing functionality of Nutch is implemented almost entirely by plugins, and is not shipped as part of the core code. This means that for every content format that the crawler must be able to handle, a new plugin must be either found or authored. Parsing plugins are available for common document formats including HTML, PDF and DOC. Nutch also enables charset detection and processing.

The interface between the crawler and searcher components of Nutch is the index. These two components are designed to be highly decoupled. However, in practice this is not quite the case. As the content of each web page is not stored directly in the index, the searcher component requires access to the crawl segments stored in the WebDB in order to produce page summaries and to provide access to cached pages.

3.9.2 Swish-e

Simple Web Indexing System for Humans – Enhanced (Swish-e) [Swish-e] is a free, open-source web crawling and indexing system. It was developed by the UC Berkeley Library in 1996, building upon the Swish web crawler. Swish was created in 1995 at Enterprise Information Technologies (EIT) by Kevin Hughes [Rabinowitz 03]. Swish-e can be used to crawl web sites, text files, mailing list archives and relational databases. Swish-e is well documented and undergoes active development and bug fixing. There is a lively mailing list that addresses issues and bugs in the system and receives regular input and feedback from project members as well as experienced users.

Swish-e has two drawbacks that could impede its effectiveness at content analysis and retrieval on the open WWW. Firstly, the crawler does not provide multibyte support. This means that the crawler can only process 8-bit ASCII code and does not support 8-bit UCS/Unicode Transformation Format (UTF-8). UTF-8 is a variable length character encoding format for Unicode. Secondly, Swish-e is designed for, and functions most efficiently on, small to medium sized data collections of less than a million documents. This does not provide sufficient scalability to conduct crawls on the open WWW.

Using the GNOME [GNOME] libxml2 [libxml2] parser and a collection of filters, the system can index standard data formats such as HTML, XML, DOC, PDF, PS and PPT among others. Indexing can be limited to selected metadata tags or XML elements, sections of a web site, or to relevant pages in a web crawl using regular expressions to decide relevancy. The default Query-Document similarity ranking is conducted using basic *tf* with the ability to bias term weighting based upon where it occurs in the page. There is also the option to use a configurable *tf-idf* function that allows the limitation of term frequency analysis to particular sections of a document. Swish-e also enables the creation of experimental ranking schemes. A drawback of the indexing functionality is that it does not support the deletion of content from an Index. To remove a document, the entire index needs to be regenerated.

3.9.3 Nazou

More recently, an IR tool-chain for ontology-based knowledge management has been developed as part of the Nazou [Nazou] Project, a Slovakian Government-funded project to develop tools for acquisition, organisation and maintenance of knowledge. The tools which are combined as part of the tool chain all address individual tasks related to the acquisition and management of content from the WWW. The tool chain has currently been implemented to search for job offers on specific Slovakian internet sites and present these to an end-user based upon their personal needs.

The Relevant Internet Data Resource Identification (RIDAR) [Gatial & Balogh 06] tool exploits existing search engines to identify relevant material based on user-supplied search queries. Basic metadata about each resource is stored in a MySQL database. The WebCrawler tool downloads the content identified by RIDAR and passes it to the Estimate Relevance of Internet Documents (ERID) [Hluchy et al. 07] tool. ERID uses Artificial Neural Network (ANN) methods of estimation to determine if the downloaded content is relevant.

The ANN needs the manual feature reduction of a pre-defined document collection before conducting an iterative Threshold Logic Unit (TLU) training process [Gatial et al. 07]. The tool can currently only process HTML, which is then converted to plain text for further manipulation.

The Offer Extraction (ExPoS) tool processes these plain text files in an attempt to extract the job offer from the text. A further text analysis tool, Offer Separation (OSID) separates blocks of job offers from documents that contain lists of offers. The final tool in the chain is called the Ontology Based Text Annotation (Ontea) tool. This tool attempts to match terms in the plain text to an existing domain ontology for a specific employment domain.

The indexing and search functionality of Nazou is addressed by the Rich Full-Text Search (RFTS) tool. Stemming methods and Slovak-specific lemmatisers are used in the indexing phase. RTFS offers keyword-based search using a basic Boolean Model approach for query-document similarity calculation. Each query is converted to a boolean SQL statement which is used to retrieve a result set from the database.

3.10 Conclusions

With the continued growth of TEL in mainstream education, there is an ever-increasing demand for high-quality digital educational content resources. This creates a major problem for educational institutions, as the development of such resources is an extensive undertaking and an expensive process. The WWW is a significant source of digital content which is potentially valuable in the generation of educational offerings. However, these resources have yet to be scalably exploited in TEL due to problems with content discoverability and reuse. Techniques and technologies employed in Information Retrieval such as focused crawling, content indexing and retrieval models can be used to aid the discovery and aggregation of educational content sourced on the WWW. These techniques and technologies can be used in the implementation of a service which supports the discovery, classification, harvesting and delivery of educational content from open corpus sources, which is an aim of this research.

4 Design

4.1 Introduction

As discussed in the previous chapters, several issues restrict the widespread adoption of TEL in mainstream education. These issues relate to the availability of educational content [Brusilovsky & Henze 07], the technical and pedagogical complexities of content repurposing and reuse [Wade & Ashman 07], the resulting demands of content authoring and TEL offering composition as well as institutional resistance to change [Charlesworth et al. 07]. These restrictions would be somewhat alleviated if sufficient volumes and varieties of freely-available educational content were accessible for utilisation by TEL environments. This chapter aims to identify and represent the core fundamental requirements of (*i*) a service which is responsible for the discovery, classification, harvesting and delivery of content from open corpus sources and (*ii*) a demonstrative TEL application which can deliver a learner-driven, pedagogically beneficial educational experience which utilises open corpus content.

The design of these applications is focused on fulfilling the objectives and goals of this thesis. This chapter begins by detailing the influences upon the design of these systems emerging from the analysis presented in both chapters two and three. These influences are related to the pedagogical aspects of TEL design, the content utilisation approaches of TEL applications and the retrieval of educational content from open corpus sources using IR techniques and technologies.

A series of design requirements are specified for a service which enables educational content discovery, classification, harvesting and delivery from open corpus sources. These requirements are initially presented in relation to the high-level design of such a service and are subsequently disaggregated into a set of technical requirements. A proposed service architecture is then presented. The chapter continues by specifying a series of requirements for a learner-driven TEL application which can utilise content provided by the above service. A series of educational design requirements and usability guidelines are defined and a set of technical requirements which build upon these are then specified. The requirements for both these systems are based upon the influences from chapters two and three previously identified. The chapter concludes by summarising the key points discussed in each of the preceding sections.

4.2 Influences from the State of the Art

The analysis conducted in both chapters two and three influenced various aspects of the applied research which is described in this thesis. The aim of this section is to provide the reader with a summary of these influences and how they affect the core properties of the open corpus content service and TEL application developed by this research. These core properties ensure:

- The discovery, classification and harvesting of subject-specific content, from open corpus sources.
- The delivery of accessible caches of categorised content for use by TEL environments.
- The support of the learner in the execution of a pedagogically beneficial educational experience which utilises open corpus content.
- The flexibility, usability and extensibility of the system functionality.

The specific implications of the influences from chapters two and three, and how they impact upon the design of the content retrieval service and educational environment are detailed under the following headings: Educational Content Creation, Dissemination and Utilisation; Educational Content Retrieval from the WWW; and Pedagogical Influences on TEL Design.

4.2.1 Educational Content Creation, Dissemination and Utilisation

Digital content repositories have begun to accumulate educational resources and web-based publication produces enormous volumes of content.

- Yet there remains no means within TEL of centrally collating and utilising subject-specific content from each of these sources.

There currently remain insufficient volumes of content in digital content repository initiatives alone to foster wide-scale reuse. This is due to a lack of mainstream engagement cause by a combination of cultural attitudes, legal and organisational problems.

- Means of collating and accessing such content in a less formal manner could help to increase the reuse of content in education.

To utilise content aggregation standards such as RSS the user must be aware of, or search for, authoritative sources of information in that area and pro-actively subscribe to these sources.

- By extending this paradigm to include the automated discovery of information sources, a very useful tool could be created. A form of seeding for this process would allow the tool to exploit the sources which the user is aware of.

Content aggregation has promoted a shift in the way content is accessed, and as a result, authored. Web pages are now more frequently created as self-contained content resources.

- This makes page-level reuse of content without the loss of context more achievable.

Content-based recommendation systems have traditionally struggled with low precision in the quality of the resources they recommend.

- Content classification could be used as a filter to narrow the pool of potential resources available to recommender systems for a particular subject area.

Recent generations of TEL applications have modified their mechanisms of content utilisation which, in theory, allows the assimilation of information from external sources.

- Functionality must still be developed which enables the discovery, classification and harvesting of external content. Methods of content repurposing will also be necessary; however, this is considered outside the scope of this research.

4.2.2 Educational Content Retrieval from Open Corpus Sources

There are numerous techniques and technologies, developed within the IR community, which independently support the discovery, classification, harvesting and indexing of content.

- Focused crawling can generate filtered, subject-specific collections of resources.
- Indexing and retrieval models can classify a resource based upon its content and compare these classifications to a query to facilitate content discovery.

The bowtie graphical description of the WWW demonstrates why it can be difficult for web crawlers to discover and access significant portions of the web.

- A thorough seeding mechanism would assist a crawler in covering a more significant portion of the WWW and reaching as many relevant pages as possible.

In many focused crawling systems the crawl preparation and scope definition can be complex and labour intensive, making them inaccessible to the non-technical user.

- An automated method of generating exemplar subject-matter collections to train the classification process and define the scope of a crawl would prove more accessible to the non-technical educator.
- The Open Directory Project [ODP] and Virtual Library Project [VLib] are human edited, topic taxonomies which classify the content of the WWW. Such initiatives could be used in a taxonomic crawling approach to define the scope of a web crawl.

Research has found that users prefer simplistic, keyword-based user interfaces when searching for information on the WWW.

- An interface which uses an intuitive search modality is essential to maximise learner engagement and minimise resistance during the execution of an education offering.

Any service which aims to implement the discovery, classification, harvesting and delivery of educational content from open corpus sources to TEL applications, must be designed with these influences in mind. It is essential that all areas of the service utilise the most applicable techniques and technologies available, to make the tool both accessible and beneficial for the education community.

4.2.3 Pedagogical Influences on TEL Design

TEL has been used to encourage the transition from static, tutor-centric education to interactive, learner-centric experiences in which the educator can act as a facilitator.

- The TEL application designed by this research should facilitate learner-driven educational experiences in which the educator can act as a guide and support.

There are complementary features between the associationist/empiricist, cognitivist and situative approaches to learning which could potentially be used in combination. To facilitate engaging and beneficial learner-driven educational experiences, the TEL application designed by this research should support elements of each of these pedagogical approaches.

- Constructivism can be supported by giving the learner more control over the pace and direction of learning.
- To support Enquiry-Based Learning, the application should enable the exploration of educational content as an aid to the learner during the execution of a learning offering.

- Mind mapping can be used to support knowledge retention and learner engagement by utilising multiple display media and varied stimuli to present information, as described by both the cognitivist and behaviourist approaches.
- Mind mapping can also be used as a tool which complements the reflection element of constructivist approaches as it allows the learner to graphically represent their knowledge of a concept domain.

These techniques actively involve the learner in the construction of knowledge, and promote engagement in the learning experience.

Based on the influences described in this section, which emerge from the analysis conducted in chapter two, technological services and applications which attempt to utilise educational content from open corpus sources must be constructed using frameworks which provide both technical and educational support. This design duality is essential if such approaches are to be beneficial to, and adopted by, mainstream education.

4.2.4 Summary

The analysis conducted in chapter two and the state of the art review and appraisal conducted in chapter three, impacted upon the design of the educational content retrieval service and educational environment being developed by this research. This section summarised the influences emerging from this analysis based upon how they affect the core properties of each of these applications. The next section will provide the reader with a more detailed insight into the design of a service which enables the discovery, classification, harvesting and delivery of content from open corpus sources.

4.3 The Discovery, Classification, Harvesting and Delivery of Content from Open Corpus Sources

This thesis has two objectives. The first objective is to investigate, prototype and evaluate an approach for supporting the discovery, classification, harvesting and delivery of educational content from open corpus sources such as the WWW. Three of the subsequent technical goals which aim to achieve this objective are: (*i*) to investigate and specify techniques and technologies which can be used to enable the dynamic discovery, classification and harvesting of content residing on the WWW and in certain defined digital content repositories; (*ii*) to prototype an application, based on these techniques and technologies,

which enables the creation of caches of educational content, sourced from the WWW, defined by subject area; and (*iii*) to investigate and implement the technologies required to index such open corpus content so that it is searchable and accessible to learners and educators.

Based on these objectives and goals, and the influences from the state of art, several requirements for such a service, to be named the Open Corpus Content Service (OCCS), have been identified. The OCCS will be responsible for the discovery, classification, harvesting and delivery of content from open corpus sources.

The following sections describe both the high-level design requirements and the subsequent technical requirements of the OCCS. The design requirements of the OCCS dictate the sources and categories of content to be targeted by the discovery service, the methods of content harvesting to be employed and the means of subsequent access to such content. The technical requirements refer to the implementation of these high-level requirements. Both the high-level and technical architectural approaches to be implemented by the OCCS, based upon the outlined requirements, are described and illustrated below.

4.3.1 High-Level Design

The OCCS aims to harness educational content, sourced from the WWW and defined digital content repositories, and to make it available to TEL applications for use by learners and educators. To achieve this objective, the OCCS should traverse the WWW in a structured fashion, enabling the discovery and accumulation of topic-specific content. Resources which are deemed relevant should be subsequently harvested and made accessible to learners and educators. The OCCS aims to provide tailored content discoverability. It should be possible to target specific internet domains or sections of the web, limit the depth into web sites which the content discovery tool drills, and specify high quality starting points for content discovery paths which will more likely lead to quality resources. This should be enabled without limiting the potential range of the discovery process; broad coverage should be possible during content discovery.

Content discovered by the OCCS should be harvested and stored in a content archive. This archive can then be indexed and made accessible to both educators and learners through a web-based user interface. To enable ease of content discovery and exploration, a traditional

keyword-based search interface was decided to be the most appropriate interface for the OCCS. Such interfaces feel intuitive to the majority of users as they are familiar with their operation. This would add to the usability of the system.

The generation of such subject-specific caches of content requires the analysis and classification of discovered content, which dictates whether a particular resource is on-topic. It is a design strategy of this research to leverage the wealth of accumulated subject classifications contained in human-edited catalogue initiatives such as the ODP [ODP] and VLib [VLib]. These classification systems can be used to help deduce which discovered content should be included in a cache for a particular subject domain. The ODP is a hierarchical catalogue of the WWW divided by subject area, any website submitted to the directory is manually reviewed to ensure it is on-topic before being added. This provides high confidence that websites are accurate to the subject category under which they are located. In the case of the OCCS, these directories can provide positive examples of content for a subject area to aid the classification process.

The high-level design requirements of the OCCS can be summarised as follows:

1. Traverse the WWW enabling the discovery of topic-specific content.
2. Provide tailored content discoverability:
    a. Target specific domains or sections of the WWW
    b. Limit the depth to which the tool will drill
    c. Specify starting points for the content discovery process
3. Enable broad coverage during content discovery.
4. Leverage the wealth of classifications contained in human-edited catalogue initiatives
5. Harvest relevant resources
6. Make these resources accessible to learners and educators.
7. Provide a keyword-based search interface

The OCCS can produce topic-specific caches of content which can be made available to learners as they execute educational experiences in particular subject areas. Such focused content delivery and exploration can help to reduce information overload and facilitate the discovery of more relevant material with less dilution of results.

Access to content is the first step towards enabling the widespread reuse and repurposing of educational content, which TEL can help to facilitate. However, the technologies which aim to enable content reuse have not yet developed to a point where the dynamic repurposing of material can be realistically implemented by the average non-technical user. This represents an enormous problem in TEL and is very much a research strand in itself [Lane 06]. However, increasing numbers of pages on the WWW are being authored as self-contained units of information, as discussed in section 2.3.3. As a result, the reuse of open corpus resources in the OCCS should be implemented in a course-grained, page-level fashion, making harvested resources available to the learner in their entirety. The learner can then explore across resources or within an individual resource and identify the sections of content which are of use in the current circumstance. The OCCS service will not provide means by which the educator or learner can adjust the granularity of a resource or repurpose portions of its content, as this was deemed out of scope.

However, this does not diminish the potential value of the OCCS service for content reuse. On the contrary, the OCCS can act as a feeder service for future developments in content reuse and repurposing. It could be used to fuel technologies such as content slicing, dynamic learning object generation and personalised course generation. Such technologies can use the OCCS as a provider of quality, open corpus educational content, which they can subsequently analyse and manipulate for inclusion in TEL offerings or for the generation of LOs.

This section has detailed the high-level design requirements of the OCCS, a service which sources and delivers educational content to TEL systems. The following section defines a number of technical requirements for the design of the OCCS which build upon the high-level requirements detailed above.

4.3.1.1 High-Level Architecture of the OCCS

Based upon the high-level design requirements outlined in the above section, a system architecture was defined for the OCCS and is illustrated in Figure 4-2. Educators, content authors and publishers all over the globe produce content which could potentially hold great value in the generation of TEL offerings. The OCCS facilitates the discovery and harvesting of this content from the WWW and selected digital content repositories. All discovered content undergoes classification for subject domain and subject-specific caches of content are

generated. These caches can subsequently be searched during the execution of a TEL offering. Alternatively these content caches could be used as a feeder service for future content repurposing functionality within TEL applications.

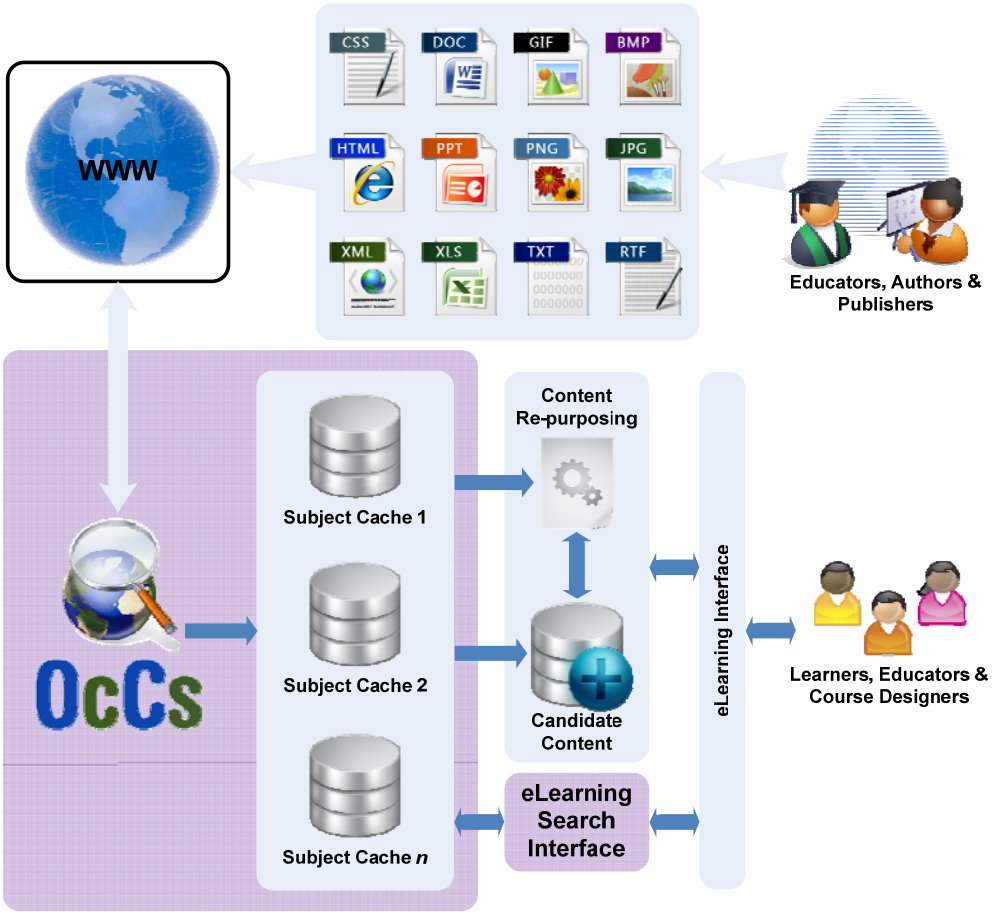


Figure 4-1 OCCS High-Level Overview

An example scenario based upon Figure 4.1 could be as follows: an educator in a third-level institution delivers a course in the department of history which covers American History and in particular examines the American Civil War. The course is supplemented by an TEL exercise which examines individual aspects of the war. The lecturer wishes to support his students through the provision of accompanying content which details all aspects of the war and in varying levels of detail. This content can then be consulted by the student as they conduct the TEL exercise. However, the educator does not want the students to consult a general purpose web search engine during the exercise, as she fears the dilution of search results with off-topic or poor quality content, and she does not want to risk students being distracted by browsing the open web. The OCCS can generate a cache of topic-specific

content for the educator, which can then be offered as a search utility within the TEL system which hosts the exercise.

4.3.2 Technical Requirements

Several core technical requirements for an effective open corpus content service have been identified. These requirements are discussed and illustrated below and expand upon both the high-level design requirements discussed in the previous section and the state of the art influences defined in section 4.2. As discussed previously, it is an approach of this research to, where possible, utilise existing open-source technologies in an attempt to avoid unnecessarily recreating technological solutions. The OCCS should employ existing open source web crawling, text classification, indexing and visualisation solutions which satisfy the technical requirements of the service detailed below.

The content discovery process of the OCCS should enable the sourcing and harvesting of content in numerous formats. Content which is of potential use in educational scenarios can exist in a variety of structural and presentation formats. Text can be wrapped in HTML, can occur in Microsoft Word DOC files or Adobe PDF files. Images can be encountered in various encoding formats such as BMP, JPEG, GIF and PNG. Video content can be MPEG or Quicktime. It is essential that the OCCS can capture and utilise all these potential educational resources.

It should be possible to specify seed sites at which a web crawl will be initiated. These sites can heavily influence the style and genre of content discovered by the service, the path which the service follows, and the potential content which the service can reach. In a non-commercial environment, such as educational web pages, high quality resources tend to link to other authoritative sources of information on that subject, see section 3.4.1. When searching for subject specific content, topical locality states that content tends to link directly to other semantically similar content, see section 3.6.1. This makes the seeding of a web crawl extremely important to the path that the crawler will follow and the content that it will thus discover. The OCCS should provide a semi-automatic seeding mechanism to support the educator in defining the scope of a web crawl. The educator should be provided with a list of potential seeds which can then be manually supplemented, reviewed or edited.

As certain web domains, and even individual web hosts, will prove to be particularly authoritative sources with regard to specific subjects, it should be possible to restrict a web crawl to defined domains or hosts. This will allow educators who have knowledge of specific sets of resources which encompass their curriculum, to collate these resources in an OCCS cache. It will also allow the restriction of a cache to certain categories of content. For instance, restricting a crawl to the .edu domain would bias the discovery of content toward course notes and other educational material. However the ability of the OCCS to crawl broad sections of the WWW in an attempt to discover quality resources from outside individual domains should not be sacrificed. The definition of crawl paths and domain limitations should be implemented as a crawl configuration option.

The OCCS should also enable the specification of depth limits on a crawl. A depth limit specifies the number of links away from the top level of a domain that the crawler is allowed to travel. This can be visualised as the number of slashes in a URL appearing after the site root, as illustrated in Figure 4-2 below.

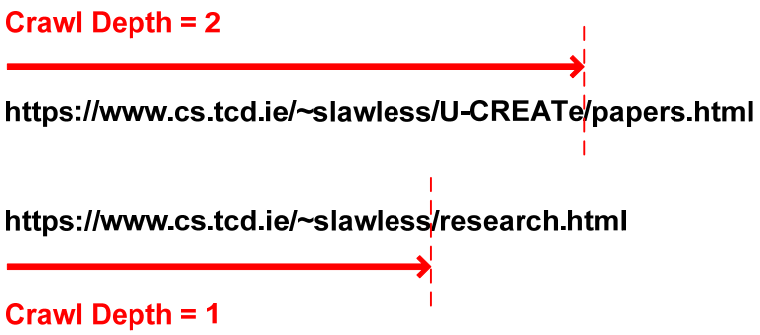


Figure 4-2 Crawl Depth

Limiting the depth of a web crawl prevents the crawler from being drawn into recursive file system structures of infinite depth, commonly referred to as “spider traps”. A spider trap is a set of web pages which intentionally, or inadvertently, cause a crawler to infinitely loop. Such traps can be created maliciously to impede crawlers or to punish poorly constructed crawlers which make unrealistic demands upon a website’s bandwidth. However, traps can also be unintentionally created by dynamic website functionality, for example, calendars which dynamically create pages from links that continually point to the next day or year. The ability to modify crawl depth is essential to avoid such spider traps and to preserve crawler performance and efficiency.

The OCCS should respect the rights and desires of content authors and owners with respect to the harvesting and use of their content from the WWW. The web crawler implemented in the OCCS should honour the Robots Exclusion Protocol, which is detailed in section 3.5.2. It is important that the OCCS discovery process be as "polite" as possible and obey all robots.txt files, not only to ensure that the wishes of content owners are respected, but also to ensure that the OCCS maintains a positive reputation amongst web server administrators.

The OCCS content discovery process should provide a configurable crawl termination process. A user may desire a specific sized content cache, and should be able to define the maximum number of content objects that they wish to be added to the cache. Upon reaching this figure the crawl should automatically complete. Alternatively the user may have a specific time period in which they can conduct the crawl. The user should thus be able to specify the desired duration of the crawl in advance, at which point the crawl should automatically complete. It should also be possible to manually monitor the crawl for a combination of these factors. When the user is satisfied that the crawl has achieved its purpose, they should be able to manually terminate the crawl. The ability to specify termination parameters will ensure the content discovery process meets the, often varied, requirements of the TEL system or individual educator.

As discussed in the previous section, a strategic design feature of the OCCS is that it should produce topic-specific caches of content. To achieve this task, it is a logical necessity that the OCCS web crawler must conduct topic-specific, rather than general purpose, crawls. This approach to traversing the WWW is termed "focused crawling". Examples of focused crawling techniques are discussed in detail in section 3.6.

To enable the "focussing" aspect of the crawl, all content discovered must be classified according to the subject domain of the required content. This implies some intrinsic knowledge of the subject area in question, on the part of the classification tool. To ensure that the classifier is correctly fine-tuned to identify content pertaining to the desired topic, a training process should be conducted in advance of each crawl. Also discussed in the previous section was the strategy of leveraging the accumulated subject categorisations of the ODP project, or similar initiatives, in the classification of discovered content. Categorisations

and categorised content from the ODP should be used in combination with other manually or semi-automatically selected positive examples of on-topic content to conduct the training process. This will support the educator in training the classifier for the scope of the crawl and not demand that they have advanced knowledge of authoritative sources of content.

In addition to categorising the discovered content by subject area, it is also necessary to categorise the content by language. The filtration of content during cache creation, based upon its language, is essential to ensure the subsequent ease of content utilisation. The languages deemed acceptable for inclusion in a specific cache should be defined in advance for each scenario. For the purposes of this research, the content cache should contain solely English language results, as the utilisation and evaluation of the content will be conducted in Trinity College Dublin, a primarily English speaking University.

The content caches generated by the OCCS focused crawler should be accessible to other OCCS processes which will conduct further analysis of the content. The content contained within each cache needs to be made accessible to the educator and learner. As discussed in the previous section, the most desirable method of enabling this discovery process is through a traditional keyword-based search interface. To enable such an interface the entire cache must be indexed to allow keyword-based search across the entire text of the resources. This indexing process must be conducted upon crawl completion and the resulting index made accessible to the user interface. Educators or learners can then use this interface to query the cache and utilise the content during the execution of TEL experiences.

This section has detailed the technical requirements of the OCCS, a service which sources and delivers educational content for use in TEL. These requirements build upon both the high-level design requirements and influences from the state of the art defined previously. The following section defines a proposed architecture of the OCCS and illustrates this architecture in relation to both the high-level requirements and subsequent technical requirements of the service.

4.3.2.1 Technical Architecture of the OCCS

Based upon the requirements defined in the previous two sections and the goals of this thesis a technical system architecture was defined for the OCCS. The high-level strategic design requirements, defined and detailed in section 4.3.1, are repeated below. The OCCS should:

1. Traverse the WWW enabling the discovery of topic-specific content.
2. Provide tailored content discoverability:
    a. Target specific domains or sections of the WWW
    b. Limit the depth to which the tool will drill
    c. Specify starting points for the content discovery process
3. Enable broad coverage during content discovery.
4. Leverage the wealth of classifications contained in human-edited catalogue initiatives
5. Harvest relevant resources
6. Make these resources accessible to learners and educators.
7. Provide a keyword-based search interface

These high-level requirements of the OCCS service were expanded upon in the section above and a set of technical requirements were described and defined. These technical requirements are:

i. The OCCS should employ existing, open source web crawling techniques and technologies to execute the content discovery process.

ii. This content discovery process should enable the identification and harvesting of content in various structural formats.

iii. A semi-automatic seeding mechanism should be implemented for generating the start points of each specific web crawl.

iv. A crawl configuration option should be implemented which enables the restriction of a web crawl to a defined set of domains or hosts.

v. A crawl configuration option should be implemented which enables the specification of depth limits on a web crawl.

vi. The OCCS should respect the rights and desires of all content authors and site owners by obeying all robots.txt files.

vii. The OCCS should provide a configurable crawl termination process.

viii. Web crawls conducted by the OCCS should be focused rather than general purpose. All discovered content should undergo language filtration and text classification for relevancy assessment.

ix. Content from the ODP should be combined with manually and semi-automatically selected content to train the text classification tool for the scope of each web crawl.

x. The generated content cache should be indexed to enable keyword-based searching across the full text of the resource.

xi. This index should be exposed via a web-based search interface.

The OCCS is designed to implement individual components, combined in a novel tool-chain. Each of the individual components addresses one of the disparate tasks which are necessary to satisfy the technical requirements of the service. This architectural design is illustrated in Figure 4.3 below.

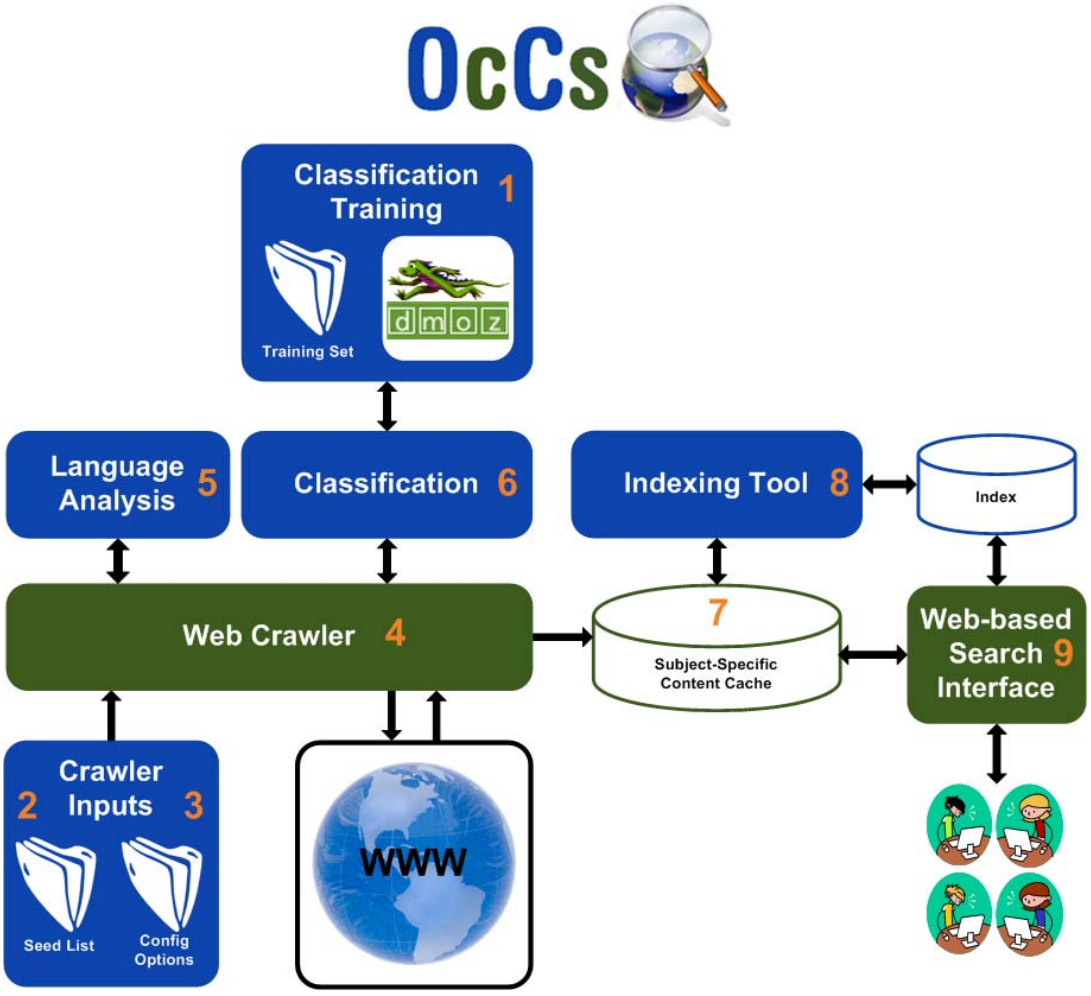


Figure 4-3 OCCS Technical Architecture

In advance of each web crawl the classification component (1) needs to be trained for the subject of the crawl. This involved the semi-automatic generation of exemplar web pages and

the utilisation of content from the relevant categories of the ODP. Once the training process is complete, a semi-automatically generated seed list (2) is used to provide start points for the crawl. A configuration file (3) is also provided which details the domains, hosts, crawl depth, termination parameters and other such options.

The web crawler (4) then traverses the WWW starting from the set of seed URLs and downloads any content it encounters. This content is analysed and filtered based upon its estimated language (5). Any content which meets the language requirements of the crawl in question should be passed along the tool-chain for subject classification (6). This content is classified in comparison to the training set of data which has been used to define the subject domain. Content which is deemed relevant to the subject of the crawl should be harvested, generating a subject-specific cache of candidate learning content (7). Cached content can then be indexed (8) generating a query mechanism for the identification of content via keyword search. This index should be made accessible via a web-based search interface (9). The OCCS can be provided as a content discovery and exploration service which can be integrated into existing TEL environments.

Consider again the scenario, described in 4.2.1.1, of an educator in a third-level institution, who delivers a course in the department of history which covers American History and in particular examines the American Civil War. To generate a cache of relevant, subject-specific content, the first task required by the OCCS is for the educator to create a training set which accurately describes the subject area. To do this the educator can provide a list of web sites which she is confident accurately describe the American Civil War. The OCCS will provide means of assistance to identify on-topic content from the WWW for use in the training set. The educator can also browse the ODP for categories which encompass or are contained within the subject area. An example of such an ODP category would be “Top: Society: History: By Region: North America: United States: Wars: Civil War”. The OCCS will then extract the content from this category and use it in combination with the manually selected web sites to train the classifier.

Once the training of the classifier is complete, the OCCS will recommend a list of seed URLs at which to start the web crawl. The educator can peruse this list and make additions, edit or remove entries. The educator can also specify configuration options for the domains to be

crawled, the depth the crawler should delve into websites, the languages which are desired for the cache and the termination parameters for the crawl. These inputs are provided to the OCCS which then initiates a web crawl and begins to analyse discovered content. Upon crawl completion a cache of content is generated which is subsequently indexed to make it accessible. Once this is complete a cache of content on the American Civil War is available to the educator for use in their TEL offering. This can be made available to the student as a free-test search interface, integrated in their TEL exercise.

4.3.3 Summary

The previous three sections of this chapter have combined to describe the design of the OCCS, a service which sources and delivers educational content to TEL systems. Section 4.3.1 detailed the high-level design and requirements of the OCCS, section 4.3.2 defined a set of technical requirements for the service which built upon the previous high-level requirements. A proposed architecture of the OCCS was defined and illustrated in relation to both the high-level and technical requirements of the service. An example scenario was also presented to demonstrate the process flow of the service using this architecture.

4.4 Open Corpus Content Utilisation in Learner-Driven TEL

The second objective of this thesis is to design, prototype and evaluate a TEL application which enables the incorporation and resource-level reuse of content from a cache generated by the OCCS during the execution of a pedagogically meaningful educational experience. This application is used to demonstrate and validate that it is possible to reuse open corpus content in a pedagogically meaningful way. Two of the subsequent technical goals which aim to achieve this objective are: (*i*) to identify the learning theories and technologies required to support pedagogically beneficial TEL experiences in an interactive, learner-driven TEL application; and (*ii*) to prototype a demonstrator educational application which enables the learner-driven exploration and resource-level reuse of content provided by the open corpus content service.

Based on these objectives and goals, and the influences from chapters two and three, several requirements, both educational and technical, for such a TEL application have been identified and will be discussed in this section. This interface is to be called the User-driven Content Retrieval, Exploration and Assembly Toolkit for eLearning (U-CREATe). U-CREATe will be responsible for delivering a learner-driven educational experience which is supported by

caches of educational content provided by the OCCS. The learner can utilise this content during the execution of the educational process.

The following sections describe the educational, technical and usability requirements of U-CREATe. The educational requirements dictate the techniques and theories which should be supported to ensure the pedagogical effectiveness of the interface. The technical requirements refer to the implementation of an interface which can facilitate offerings which support these techniques and theories. The usability requirements ensure that the interface is intuitive and effective. These requirements are described and illustrated below.

4.4.1 Educational Requirements

This section describes the educational requirements for the U-CREATe application. These requirements ensure that the educational experiences offered via U-CREATe are engaging and pedagogically beneficial to the learner. U-CREATe should deliver educational offerings which actively support elements of specific educational theories and techniques. In contrast to the OCCS, U-CREATe must be primarily pedagogically rather than technically focused.

Based on the analysis presented in chapter two and the influences detailed in section 4.2.3, U-CREATe should facilitate learner-driven educational methods, based on active learning strategies, which directly involve learners in the knowledge acquisition process, and facilitate their progress through the learning objective hierarchy.

To facilitate engaging and beneficial learner-driven educational experiences U-CREATe should be designed to supports elements of each of the influencing pedagogical approaches discussed in section 2.2. The educational theory of constructivism is widely regarded as a means of supporting pedagogically effective, learner-driven education, both within traditional classroom-based scenarios and in interactive technology enhanced learning offerings. Constructivism can be supported by ensuring that the learner has control over the pace and direction of learning offerings conducted in U-CREATe. Enquiry-Based Learning promotes the acquisition of problem solving and research skills by the learner. Enquiry-Based Learning promotes using and consuming educational content to aid the development of such information-processing and problem-solving skills. This approach can be facilitated by U-CREATe enabling the exploration of educational content as an aid to the learner during the execution of a learning offering. Both of these techniques actively involve the learner in the

construction of knowledge, cultivate a sense of ownership of the learning process and promote engagement in the learning experience.

Mind mapping, as detailed in section 2.2.1.2 and Appendix A, is a graphical technique which U-CREATe can utilise to support knowledge retention during the learning process. Mind maps employ multiple display media, such as graphical visualisations combined with text and images, to present information. The use of such varying display methods can aid the encoding and retention of knowledge as described by the cognitivist approach to education. Mind mapping can also aid learner engagement in the learning offering as it provides varied stimuli, an influence from the behaviourist approach.

Mind mapping can also be used in U-CREATe to complement constructivist methods, through the construction and representation of conceptual knowledge models and facilitating learner reflection upon these models. The creation of a mind map allows the learner to graphically represent and visualise their knowledge of a subject domain and use this to reflect and subsequently adapt or build upon this knowledge model. Constructivism asserts that prior knowledge is used as a framework for knowledge acquisition. The process of reflection can also help to improve knowledge retention. In essence, how an individual interprets and comprehends information can influence how and what they learn. Mind mapping can formalise this relationship between a learner's knowledge and the concepts being studied.

As the learner develops their representation of a subject area in U-CREATe, individual concepts within the subject area or relationships between concepts can be explored further through searching for and consuming complementary educational content provided by the OCCS. This can support the acquisition and development of knowledge in the areas which the learners themselves feel demand the most attention. U-CREATe should be used to supplement learning conducted through traditional tutor-driven scenarios. Through self reflection and concept exploration the learner can conduct the exercise in a manner which suits both their knowledge level and preferred learning methods.

It is critically important that educators do not feel disenfranchised from the learning experiences which U-CREATe offers. U-CREATe is by no means intended to replace the educator at any point of the educational process, it should more accurately be considered as a

support to the traditional, classroom-based means of teaching. The educator should remain active at all stages of the educational process, from design through to execution and analysis of the experience. During the creation of educational experiences, a key source of information is the educator's knowledge of the subject area. However, as U-CREATe supplements the educational process with the delivery of interactive learning offerings, the educator's role will be transformed from the traditional position as a disseminator of information to a more interactive facilitator of knowledge acquisition [Ausubel 00] [Novak & Canas 04].

The educational requirements of U-CREATe can be summarised as follows:

1. Facilitate learner-driven educational methods which directly involve learners in the knowledge acquisition process.
2. Support Constructivism by ensuring that the learner has control over the pace and direction of the learning offering.
3. Support Enquiry-Based Learning by enabling the exploration of educational content as an aid to the learner during the execution of a learning offering.
4. Utilise mind mapping techniques to support knowledge retention during the learning process and to complement the Constructivist approach.
5. Act as a supplement to learning conducted through traditional tutor-driven scenarios through the provision of a tool which promotes reflection and knowledge refinement.

It should be noted that it is not the author's contention that the methods implemented by U-CREATe are the only means of utilising caches of content created by the OCCS. On the contrary, it is the hope that many diverse and novel educational applications will be able to utilise such content. U-CREATe is used to demonstrate and validate that it is possible to reuse open corpus content in a pedagogically meaningful way.

This section defined a set of educational requirements for the design of U-CREATe, a learner-driven TEL application which facilitates the generation of graphical representations of a learners knowledge models using mind maps. U-CREATe also facilitates the exploration of caches of educational content aggregated by the OCCS. Through a mixture of pedagogy, subject matter expertise and facilitating technologies, better informed TEL applications can

be developed to actively involve and support the learner during the learning process. The following section defines a number of technical requirements of U-CREATe which build upon these educational requirements.

4.4.2 Technical Requirements

Based on the educational requirements defined above and influences from chapters two and three, core technical requirements for an effective learner-driven TEL system called U-CREATe have been identified and are detailed below.

To enable U-CREATe to deliver a learner-driven educational offering which supports the generation of mind maps, a graphical user interface (GUI) should be implemented. In line with the strategic approach of this research, U-CREATe should utilise and extend existing open source solutions, where possible, during the implementation of this GUI. To support the learner during the generation of a mind map, caches of complementary content, which the learner can search and explore, are required. Such caches of topic-specific content should be provided for each educator-defined scenario to enable the learner to investigate concepts and enhance their knowledge models of a subject area.

These content caches should be provided by the OCCS and the OCCS search functionality should be accessible via the U-CREATe GUI. The learner should be able to browse and explore selected content in a separate window without navigating away from their mind map. This will enable the learner to consult related content and investigate individual concepts as a map is developed. A learner should be able to tag each node in a mind map with links to specific pieces of content from the OCCS cache which were informative or complementary in the exploration of that concept. These tags should be hyperlinks to the content in question to allow the learner to refer back to each piece of content as they browse the mind map.

The technical requirements of U-CREATe can be summarised as follows:

- **i.** Implement a Graphical User Interface which supports the generation of mind maps.
- **ii.** Utilise and extend existing open source solutions during the GUI implementation.
- **iii.** Caches of complementary contents should be provided to support the learner during the authoring of a mind map.

iv. These content caches should be provided by the OCCS and the OCCS search functionality should be accessible via the U-CREATe GUI.

v. The learner should be able to browse and explore selected content in a separate window without navigating away from their mind map.

vi. A learner should be able to tag each node in a mind map with hyperlinks to specific pieces of content from the OCCS cache.

This section defined a set of technical requirements for the design of U-CREATe, a learner-driven TEL application which facilitates the exploration of educational content from open corpus sources. These technical requirements extend the educational requirements specified in the previous section.

4.4.3 Usability Guidelines

The defined educational requirements for U-CREATe are paramount in the design of the user interface. However, to ensure the interface is both practical and engaging for the learner a series of usability guidelines have been defined which guide its design. In order to create an effective and usable system there are core principals of usability which must be adhered to [Shneiderman 98] [Dagger 06].

The diversity of the U-CREATe user base must be acknowledged [Shneiderman 98] [Hansen 71]. While it is not practical to attempt to support every user's individual requirements in a non-adaptive system, identifying and characterising a user base can significantly increase system usability. In the case of U-CREATe, for the purposes of this research, the user base is composed of undergraduate university students, pursuing courses in computer science.

Interface usability can also be significantly improved by adhering to what are termed the "eight golden rules of interface design" [Shneiderman 98]. These rules are defined as:

- Strive for Consistency
    - The interface should produce consistent sequences of actions for similar situations. Terminology should be standardised across all menus, prompts and screens. Colours, fonts etc. should also be standardised throughout the system.
- Provide Shortcuts for Experienced Users

    - As a user becomes familiar with a system their desire to increase the speed of interaction increases. This can be achieved by reducing the number of interactions necessary through the provision of function keys and hidden commands.
- Offer Informative Feedback
    - Every user action should produce an informative system response. Frequent or minor actions can produce subtle responses while infrequent or major actions should produce more significant responses.
- Design Dialogues to Yield Closure
    - Informative feedback via dialogs should be used to signify the completion of action sequences. This gives the user a feeling of accomplishment and allows them to prepare for the next sequence of actions.
- Offer Error Prevention and Simple Error Handling
    - The system should be designed to limit the potential for user errors. If such errors do occur the system should detect this and offer simple, comprehensible feedback for handling the error.
- Permit Easy Reversal of Actions
    - Where feasible, provide functionality so that users can undo or redo actions automatically. This helps to avoid anxiety in users.
- Support User Empowerment
    - The system should be designed so that the user is always the initiator of an action rather than a respondent. This ensures the user feels in total control of the operation of the system.
- Reduce Short-Term Memory Load
    - The limitation of human short-term memory means that the system should provide action sequences with no more than seven plus or minus two actions. This can be achieved by ensuring displays are simple and consolidating multiple pages.

Adhering to these guidelines during the design of the U-CREATe user interface will help to ensure that the system is effective, efficient and user friendly.

4.5 Summary

This chapter discussed the design of: (*i*) the OCCS, a service which generates caches of topic-specific content from open corpus sources; and (*ii*) U-CREATe, a TEL application for the delivery of a learner-driven, pedagogically beneficial educational offering which utilises open corpus content provided by the OCCS. Core fundamental requirements for each system were defined.

The requirements for the OCCS are both high-level and technical in nature and are based upon influences emerging from the analysis conducted in chapter two and the state of the art review and appraisal conducted in chapter three. A proposed architecture for the OCCS is presented. This service will employ a focused web crawler to discover, harvest and classify content from the WWW. Harvested content will be stored in subject specific caches and indexed for keyword-based search through a web interface.

The requirements for U-CREATe are both educational and technical in nature and are based upon influences emerging from the analysis conducted in chapter two. U-CREATe is a learner-driven TEL application with a mind mapping GUI. The OCCS search interface will be accessible via U-CREATe to enable the exploration of subject specific content during the creation of a mind map. Usability guidelines were specified to ensure the design of the U-CREATe user interface is both user friendly and effective.

5 Implementation

5.1 Introduction

Chapter four of this thesis described the design of a content discovery and retrieval service for TEL and a method of exploring and integrating such open corpus content within an educational experience. These designs are based upon a set of requirements, both technical and educational. This chapter now describes the implementation of the Open Corpus Content Service (OCCS), which delivers methods of content discovery, harvesting, classification and delivery. The chapter also discusses the implementation of the User-driven Content Retrieval, Exploration and Assembly Toolkit for eLearning (U-CREATe), which delivers an interface for the learner-driven creation of Mind maps and the ability to explore and reuse educational content provided by the OCCS.

The technical implementation conducted in this research is detailed in this chapter through a series of sections focusing on the architecture of both the OCCS and U-CREATe. The individual components of both these systems are examined in detail. Described in the following section are the various methods of content discovery and retrieval supported by the OCCS. The chapter then proceeds to describe the framework for content exploration and reuse employed by U-CREATe.

5.2 The Open Corpus Content Service

The development of a content discovery and retrieval tool, which can be utilised by TEL environments, can provide an essential first step towards leveraging the vast volume of knowledge and information available from open corpus sources, such as the WWW, for use in educational offerings. The OCCS, as described in chapter 4 of this thesis, is designed to provide such functionality. A series of technical requirements were defined, which the OCCS must implement to satisfy the objectives of this thesis. These requirements are:

i. The OCCS should employ existing, open source web crawling techniques and technologies to execute the content discovery process.

ii. This content discovery process should enable the identification and harvesting of content in various structural formats.

iii. A semi-automatic seeding mechanism should be implemented for generating the start points of each specific web crawl.

iv. A crawl configuration option should be implemented which enables the restriction of a web crawl to a defined set of domains or hosts.

v. A crawl configuration option should be implemented which enables the specification of depth limits on a web crawl.

vi. The OCCS should respect the rights and desires of all content authors and site owners by obeying all robots.txt files.

vii. The OCCS should provide a configurable crawl termination process.

viii. Web crawls conducted by the OCCS should be focused rather than general purpose. All discovered content should undergo language filtration and text classification for relevancy assessment.

ix. Content from the ODP should be combined with manually and semi-automatically selected content to train the text classification tool for the scope of each web crawl.

x. The generated content cache should be indexed to enable keyword-based searching across the full text of the resource.

xi. This index should be exposed via a web-based search interface.

Each of the components of the OCCS has been implemented to meet one or more of these requirements. An summary of the OCCS tool chain is provided in the next section. This is followed by a detailed description of the various components of the OCCS and the means by which they were selected and implemented.

5.2.1 Tool Chain Summary

The OCCS tool chain enables the discovery, classification, harvesting and indexing of content from open corpus sources. It can be used to generate subject-specific caches of content for use in TEL. The OCCS uses a focused web crawler, developed using Heritrix [Heritrix], the Internet Archive's open source web crawler, Rainbow [Rainbow], a statistical text classification tool and JTCL [JTCL], a language guesser. Crawls are conducted which target content on the WWW based upon an educator defined scope. Such crawls are an incremental process in which a URI is selected from those scheduled and the content at that URI is fetched and classified for relevance to scope.

Content classification involves filtering by language and conducting a comparison between the resource in question and a subject classification model. Rainbow must be trained in advance of each crawl to generate the classification model based upon the scope defined by the educator. A combination of keyword files and ODP categories are used to generate positive and negative training sets of content. The classifier uses these training sets to build a statistical model of the subject area. The OCCS then uses this model to ascertain the relevancy of crawled content to the scope of the crawl.

To generate an index of the stored content, the OCCS incorporates NutchWAX [NutchWAX]. NutchWAX sequentially imports, parses and indexes the entire collection of content in the OCCS cache. Wera [Wera] is used in the OCCS to link the NutchWAX index with the content cache and allow the visualisation of the archived content.

5.2.2 Components of the OCCS

Requirement **i** specifies that the implementation of the OCCS should be based upon an open source web crawling solution. Each of the crawlers detailed in the state of the art analysis in section 3.7 satisfy various aspects of the feature set required by this research. However none provide an ideal solution independently, which is to implement the discovery, classification and harvesting of content from the open WWW in a subject-specific fashion. Requirement **ii** is that the crawling solution implemented should be able to harvest content in a variety of structural formats.

Swish-e [Swish-e] is a stable platform that undergoes regular development and bug fixing. It can handle the majority of document types and can perform metadata extraction. However, the crawler is designed for small to medium crawls and as this research is conducting crawls across the WWW, this is not sufficient. Swish-e also has no multibyte support and indexes created using it indexer cannot have records deleted from them.

Combine [Combine] has great promise as a focused crawler and could produce a very efficient implementation, as it can synchronously run multiple crawler instances. However, it requires a lot of in depth pre-crawl work to create subject descriptions and term weightings which guide the progress of each crawl. At the time of crawler selection, Combine was also still in a development cycle as part of the ALVIS project [Ardö 05] and the focused version of the crawler was not considered stable enough to be adopted.

The iVia Nalanda [Nalanda] crawler can deliver very accurate focusing of crawls and is very efficient in space and memory usage. It assesses what links are the most likely to lead to relevant information, and can thus prioritise the links to follow. iVia uses a canonical topic taxonomy to define the scope of each crawl. This combines a canonical classification tree and a corresponding set of example pages. This set of exemplar pages must be manually provided. However, there is no assistance provided in selecting these pages. The granularity of the taxonomy must also be set and refined, quite an intensive process. As the OCCS will be utilised by educators in the course of their work, it is undesirable to have demanding tasks during crawl preparation. It would be preferable if they could define the scope of the crawl as simply as possible and be supported with the semi-automatic selection of exemplary content. During the crawler review and selection process attempts were made to contact the author of iVia without success.

Nazou [Nazou] is a more recently developed IR tool-chain which provides focused crawling functionality through the combination of the RIDAR, ERID and WebCrawler tools. RIDAR utilises existing search engines to identify relevant sites which are passed to the WebCrawler. The focussing aspect of the crawl is conducted by ERID which uses an Artificial Neural Network based on the Threshold Logic Unit approach. This approach to content classification requires a complex feature set definition and training process which would prove difficult for a non-technical user. The tool-chain shows great promise in generating subject-specific collections of content. However, these tools have only recently been developed and have currently been evaluated using very small collections of content, numbering in the hundreds, from three specific Slovak job sites.

Nutch [Nutch] supports general purpose crawling and has built-in indexing functionality. It undergoes regular development and bug fixing by an active community of users. Nutch can support the parsing of basic content formats such as HTML, PDF and DOC. However, to parse any other content formats for link extraction a custom plugin must be authored. Nutch version 7.0, which was the current release at the time of review, was also quite complex to configure. This seems to have been improved upon by later versions of the software.

The Heritrix [Mohr et al. 2004] crawling solution follows a standard, rather than focused, crawling methodology. However, it is designed using a pluggable architecture which allows for the easy addition of extra components. Additional features such as the classification of the text of each web page can be added as a component in the processor chain to dictate what content is harvested and turn Heritrix into a focused crawler. Heritrix is supported by regular bug fixes and an active developer network and mailing list. It can handle the majority of content formats and can extract links from, not only HTML, but javascript, flash and other formats. Heritrix also satisfies requirements **iv**, **v**, **vi**, **vii**. It provides configuration options which dictate crawl domain, crawl depth, politeness policy and termination process. It was decided that Heritrix was the most stable, well supported and easily configurable of the crawling solutions reviewed and this was the crawler implemented in the OCCS tool chain. More detail on Heritrix and its architecture can be found in section 5.2.3.

Requirement **iii** is for a semi-automatic seeding mechanism to support the educator during crawl preparation. As discussed in section 4.4.2, the seeding of a web crawl is an extremely important process. Each crawl should be provided with a number of authoritative web sites in the subject domain of the crawl from which to begin its web traversal. As detailed in Appendix B, Google is by far the most popular web-scale search engine, which would indicate that it gives the most reliable and accurate results for general web searches. If the educator provides a list of keywords which describe the subject area of the crawl to be performed by the OCCS, Google can be utilised to identify the most authoritative page for each keyword. These pages can then be reviewed by the educator and used as seeds for the crawl. As the educator's knowledge is paramount when defining the subject area in question, it should be possible to manually supplement this list if they see fit. More detail can be found on the OCCS seeding mechanism in sections 5.2.9.

As Heritrix is a general purpose web crawler, and requirement **viii** states that crawls conducted by the OCCS should be focused, a method of content classification that could be plugged into the Heritrix processor chain was required. Rainbow [Rainbow] is a statistical text classification tool which offers deployment as a server. This means that as content is discovered by Heritrix, it can be passed to the Rainbow server for classification and a decision can made with regard to its relevance before continuing with a crawl. Rainbow needs to be trained in advance of each crawl for the subject area in question. This requires the

provision of exemplar pages. As mentioned above, it is undesirable for the educator to be required to manually source these pages. As a result, the seeding mechanism of the OCCS was extended to provide exemplar web pages which are used during the classification training process. This was combined with exemplar content from the ODP, to satisfy requirement **ix**. This meant that the educator is only required to provide a keyword list for the subject area and a list of the relevant categories from the ODP. Exemplar pages for each of these keywords and categories are then automatically generated. More detail on Rainbow and the training process can be found in sections 5.2.4 and 5.2.9.

A second part of requirement **viii** is that the web crawling solution provided by the OCCS should contain a language identification and filtration process. N-gram based text classification [Cavnar & Trenkle 94] has been found to be up to 100% accurate at identifying the language of written text when applied to documents longer than 21 words in length [Capstick et al. 00]. TextCat [TextCat] is an implementation of an n-gram based text categorisation algorithm [Cavnar & Trenkle 94] which can be used to identify the language of a page of text. A component is plugged into the Heritrix processor chain in a similar fashion to Rainbow. As Heritrix is implemented in Java it was decided to use JTCL [JTCL], a java implementation of TextCat. Any content discovered by Heritrix can be initially passed for language identification. If the language of the content is English then it can be passed to the Rainbow server for classification. More detail on JTCL can be found in section 5.2.5.

Requirement **x** is that caches of content which are generated by the crawling process of the OCCS must be indexed so that learners can search across the full text of the content collection and discover content relevant to specific queries. Six individual indexing solutions were examined in the state of the art chapter in section 3.8.5.

ht://Dig [ht://Dig] is an indexing solution which uses fuzzy word indexing algorithms to identify term occurrence in a collection of content based upon criteria such as synonym rules and word ending rules. Keyword queries are then matched based upon term weighting. ht://Dig is not suited to implementation in the OCCS as the system is specifically designed for the search requirements of a single company, campus or web site. It does not support the web-scale indexing and searching which will be needed for this research.

Swish-e [Swish-e], as described above, is a web crawling and indexing solution which provides support for the majority of content formats. Query-document similarity rating is conducted using term frequency or *tf-idf* measures. However, once generated a Swish-e index cannot have a record deleted, the entire index must be regenerated. Swish-e also functions most efficiently on small to medium-sized content collections of less than a one million objects.

Xapian [Xapian] is an indexing and search solution implemented in C++. Xapian is highly adaptable and its query-document similarity rating is conducted using the Probabilistic IR model. Boolean term weighting in the default method in the system, although it does allow the definition of custom weighting schemes. WebGlimpse [WebGlimpse] is indexing and search software which runs on Unix. Glimpse in the indexing component of the software. It uses term frequency weighting and the search interface used boolean matching and pattern matching methods.

Lucene [Lucene] is a full-text indexing and search solution implemented in Java. Lucene supports *tf-idf* style term weighting and a combination of Boolean and VSM query-document similarity rating. Nutch [Nutch] is built upon Lucene and uses its indexing methods. Nutch supplements this with parsing tools for other content formats such as HTML, DOC and PDF. It also adds web specific weighting methods which exploit the standardised structure of web-based content. It was decided that Nutch was the most feature rich and web-specific of the indexing tools examined. However, Heritrix had been selected as the web crawling solution to be implemented in the OCCS which produces archives of content in the ARC file format which Nutch does not support. NutchWAX [NutchWAX] stands for Nutch with Web Archive eXtensions and is an implementation of Nutch with added support for ARC web archives. NutchWAX is the indexing solution implemented in the OCCS. More detail on Lucene, Nutch and NutchWAX can be found in section 5.2.6.

The final requirement emerging from the design of the OCCS is requirement **xi**. It states that there should be a web-based search interface available through which learners can conduct searches across the index to discover content within the OCCS cache. As Heritrix was selected as the crawling component of the OCCS and NutchWAX selected for indexing, it was essential that the search interface software was compatible with the NutchWAX index

and could extract content from the web archive created by Heritrix. WERA [WERA], which stands for Web Archive Access, is specifically designed to function with web archives and is compatible with NutchWAX indexes. WERA is implemented in PHP and provides keyword-based search functionality. More detail on WERA can be found in section 5.2.7.

This section outlined the individual components of the OCCS and how each of these components satisfy the various technical requirements for the service which were defined in chapter 4. The next section describes how these components fit into the architecture of the OCCS.

5.2.3 Architecture of the OCCS

The architecture of the OCCS, as illustrated in Figure 5-1, facilitates the discovery, classification, harvesting, indexing and delivery of content from open corpus sources. The OCCS achieves this by implementing various components which address each of these tasks.

A focused web crawler conducts traversals of the WWW, analysing the content it encounters and categorising by topic. Content deemed relevant to the subject of the crawl is harvested, generating subject specific caches of candidate learning content. The web crawler implemented in the OCCS, Heritrix [Heritrix], is combined with a language guesser, JTCL [JTCL], and a statistical text classifier called Rainbow [Rainbow] to create a focused, rather than general purpose, crawler. JTCL and Rainbow govern what content is harvested and added to the content cache and what content is disregarded.

Cached content is then indexed and made searchable via a web interface. An internet archive indexing tool, NutchWAX [NutchWAX] is used to index the caches of content generated by the crawler. WERA [WERA] is a web archive collection visualisation tool that is used in the OCCS to search the content caches and browse selected resources. The OCCS is provided as an autonomous service that can potentially replace the current method of content sourcing in TEL environments.

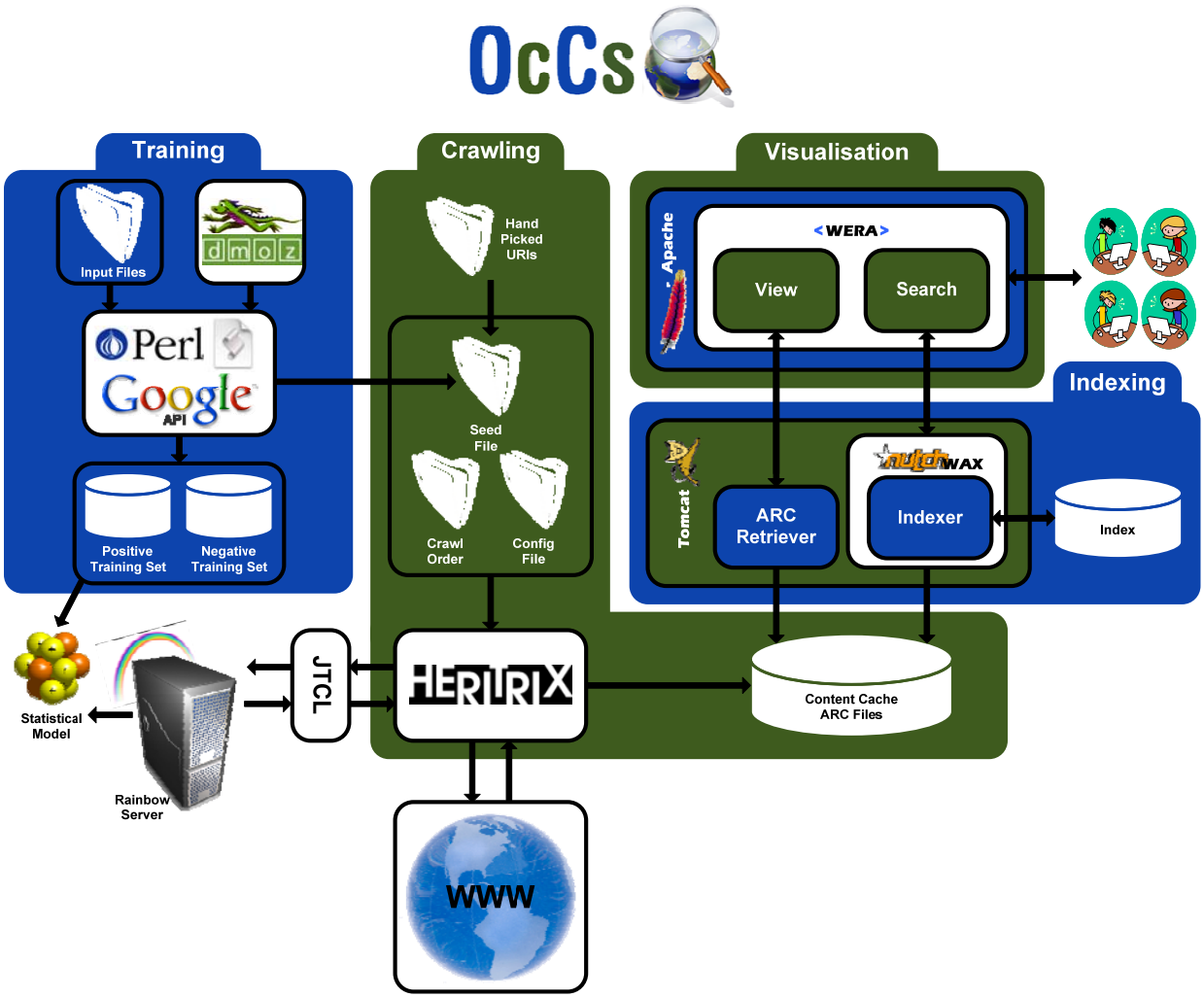


Figure 5-1 OCCS Architecture

The previous two sections have detailed the selection of the individual components of the OCCS and how these components fit into the system architecture. The following sections will provide more detail on each of the open source tools which were selected and implemented as part of the OCCS tool chain.

5.2.4 Heritrix

Heritrix [Mohr et al. 2004], as described in section 3.7.4, is an open source, extensible, web-scale, archival quality web crawler, developed by the Internet Archive [Internet Archive] and available under the GNU Lesser General Public License [LGPL]. Heritrix was first created in 2003 and has developed into a stable platform with a large development community and user base. These communities provide regular bug fixes and development assistance is provided by a lively mailing list. The crawler is implemented in Java and can be used to implement numerous crawling strategies. These strategies are detail below.

5.2.4.1 Crawling Strategies

Broad crawls are defined as "large, high-bandwith crawls in which the number of sites and the number of valuable individual pages collected is as important as the completeness with which any one site is covered. At the extreme, a broad crawl tries to sample as much of the web as possible given the time, bandwidth and storage resources available."

Focused crawls are defined as "small to medium-sized crawls, usually less than 10 million documents, in which the quality criterion is complete coverage of some selected sites or topics."

Continuous crawls are defined as "crawls that revisit previously fetched pages, looking for changes, as well as discovering and fetching new pages, even adapting its rate of visitation based on operator parameters and estimated change frequencies."

Experimental crawls are defined as "for use by groups who want to experiment with crawling techniques in areas such as choice of what to crawl, order in which resources are crawled, crawling using diverse protocols, and analysis and archiving of crawl results."

5.2.4.2 Heritrix Architecture

The Heritrix architecture, illustrated in Figure 5-2, is designed to provide a generic crawling framework into which various interchangeable components can be plugged to vary the crawling strategy and support the development of new crawler features. In the OCCS, Rainbow and JTCL are plugged into the Heritrix framework to implement a focused crawling strategy. Heritrix executes crawls in a similar recursive fashion to the majority of web crawlers.

- A URI is chosen from among those scheduled.
- The URI is fetched and analysed.
- The results are archived.
- Any discovered URIs of interest are added to the schedule.
- The URI is marked as complete and the process is then repeated.

The most important components of Heritrix are the CrawlController, Scope, Frontier, ToeThreads and Processor Chain.

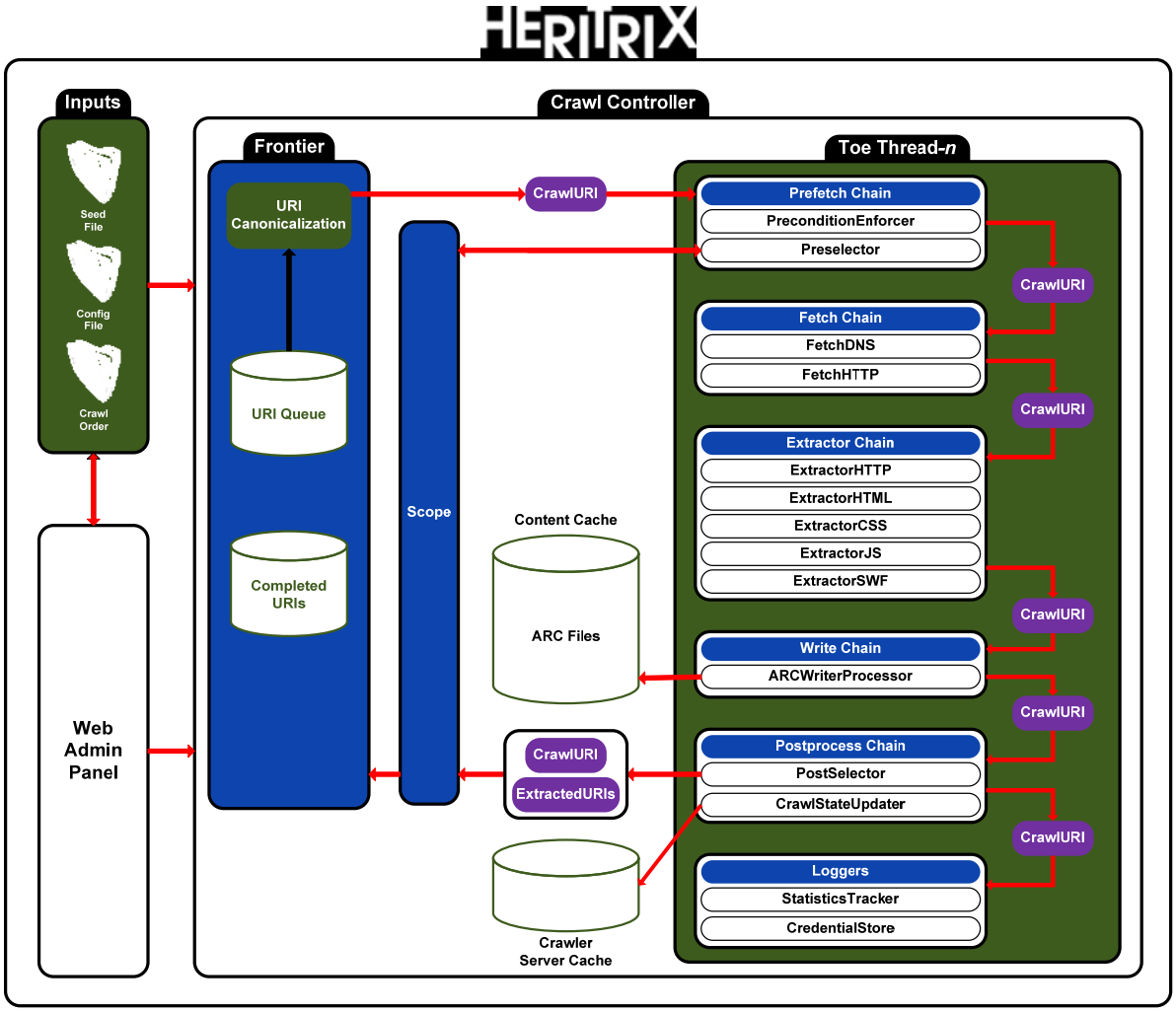


Figure 5-2 Heritrix Architecture

CrawlController and ToeThreads

The CrawlController, often referred to as the "Global Context" of the crawler, manages all the Heritrix components which cooperate to conduct a web crawl and provides a high-level interface to any crawls currently in progress. Subcomponents of Heritrix communicate with each other through the CrawlController. The Heritrix crawler is multi-threaded to enable the sequential processing of many URIs during network and server I/O intervals. Each worker thread is called a *ToeThread*. When crawling, Heritrix employs a configurable number of ToeThreads to process each URI and the CrawlController acts as a master thread to each of these ToeThreads. Each ToeThread is provided with a URI, which the CrawlController requests from the Frontier. Each thread applies all of the Processors defined in the Processor Chain to the web document retrieved from the URI and reports back to the Frontier upon completion of the Processor Chain. The ToeThreads are thus named as the Internet Archive's vision for their web crawler was not the traditional image of a spider traversing a web, but

more like a centipede or millipede: fast and many-segmented. They state “Anything that crawls over many things at once would presumably have a lot of feet and toes. Heritrix will often use many hundreds of worker threads to crawl, but 'WorkerThread' or 'CrawlThread' seemed mundane. So instead, we have ToeThreads.”

Scope

The Scope determines what URIs are relevant to a particular crawl. This includes the seed URIs used to initiate a crawl, plus the rules by which it is determined if a discovered URI is relevant to the crawl underway. There are several versions of the scope that are available for use within Heritrix.

BroadScope provides the ability to limit the depth of a crawl. This means the number of links away from the top level of a domain that the crawler is allowed to travel. BroadScope does not impose any limits on the hosts, domains or URI paths that the crawler is allowed to process.

SurtPrefixScope allows for the specification of defined domains, individual hosts or path-defined areas of hosts that the crawler is permitted to process. SurtPrefixScope considers whether any URI is inside the primary focus of the scope by converting the URI to its Sort-friendly URI Reordering Transform (SURT) form [SURT]. SURT is defined as “a transformation applied to URIs which makes their left-to-right representation better match the natural hierarchy of domain names.” SurtPrefixScope then assesses this SURT form to ascertain if it begins with any of a number of SURT prefixes [SURT Prefix]. SURT prefixes are a shared prefix string of all SURT form URIs in the same 'area' of interest for web crawling. A collection of sorted SURT prefixes is an efficient way to specify a crawl scope i.e. any URI whose SURT form starts with “*n*-prefix” should be included. The set of SURT prefixes deemed within the scope of the crawl can be specified through the extraction of SURT prefixes from the supplied seed URIs, or through the provision of an external file containing a defined list of SURT prefixes, or both.

FilterScope is highly configurable and as a result can produce a wide variety of behaviour. A number of filter modules are provided in Heritrix and these can be combined and sequenced

using the FilterScope to define the required crawl scope. Some of the more common Heritrix filters are described below:

- OrFilter: Combines *n* Heritrix filters and performs a logical OR on them, returning true if any of the filter conditions are met.
- URIRegExpFilter: Returns true if a URI matches a pre-defined regular expression.
- ContentTypeRegExpFilter: Compares a pre-defined regular expression to the response Content-Type header. Returns true if the content type matches the regular expression. ContentType regexp filter cannot be used until after fetcher processors have run. Only then is the Content-Type of the response known.
- SurtPrefixFilter: Returns true if a URI is prefixed by one of the SURT prefixes supplied by an external file or deduced from the seed set of URIs.
- FilePatternFilter: Compares the suffix of a passed URI against a pre-defined regular expression pattern and returns true if a match occurs.
- PathDepthFilter: Examines the CrawlURI path depth and compares it to the max-path-depth value of the crawl. Returns true if the CrawlURI path depth is less than or equal to the maximum depth allowed.
- PathologicalPathFilter: Checks if a URI contains a repeated pattern. This check is performed in an attempt to avoid crawler traps where the server repeatedly adds the same pattern to the requested URI. For example;
  http://host/img/img/img/img....
  The filter returns true if such a pattern is encountered.
- HopsFilter: Returns true for all URIs passed in with a Link hop count greater than the max-link-hops value of the crawl.

For the purposes of this research, BroadScope was the scope implemented in the OCCS. When a crawl is initiated, the OCCS does not have a list of defined hosts or domains within which the crawler should remain, and the crawler's discovery path should not be restricted. Once the content of a site is considered to be relevant by the classification step then it is included within the scope of the crawl. Evaluations [Cho et al. 98] [Micarelli & Gasparetti 07], detailed in section 3.6.2.1, showed that if the goal of a crawl is the discovery of pages relevant to a specific topic, then breadth-first crawling strategies perform most successfully. This can be attributed to the fact that high quality pages in a given subject area often refer to other authoritative pages in the same area [Kleinberg 99b] [Micarelli & Gasparetti 07]. The

use of BroadScope also complements link proximity theory. As the limits of a breadth-first focused crawl are smaller than a general purpose crawl, this allows the crawler to revisit and dig deeper into sites which have been deemed relevant.

Frontier

The Frontier maintains the internal state of the crawl by tracking which URIs are scheduled for crawling, and which URIs have already been processed and completed. It is responsible for the selection of the next URI to be crawled and prevents unnecessary re-crawling of completed URIs, unless of course re-crawling is a feature of the crawling strategy. It achieves this by maintaining a series of URI queues. The default Frontier implementation of Heritrix offers a breadth-first, order of discovery policy for choosing URIs to process, however there are various Frontier options available.

BdbFrontier is the default in Heritrix. It crawls the URIs in the schedule queue in a breadth-first, order of discovery manner. Configuration options are available to control how Heritrix throttles activity against particular hosts and whether there is a bias towards finishing hosts in progress, also called site-first crawling, or cycling between all hosts with pending URIs. Any discovered URIs are only crawled once, however the DNS information for each URI can be updated at scheduled intervals.

DomainSensitiveFrontier is a subclass of the BdbFrontier which allows the specification of an upper-bound on the number of documents downloaded per-site. This is achieved by exploiting an override option in Heritrix which allows individual settings to be overridden on a per-domain basis. A filter is added to block further fetching once the crawler has reached the specified site limits.

AdaptiveRevisitingFrontier recursively visits all URIs in the scheduled queue, both discovered and seed. The crawl interval is configurable and can be altered mid-crawl based on the perceived update frequencies of sites. Alternate intervals can also be specified based upon the document's MIME type [MIME].

For the purposes of this research, the OCCS utilised the BdbFrontier frontier option, again, as the crawler's possible paths to specific domains or hosts should not be restricted. The entire range of domains that contain on-topic content is not available in advance of a crawl, and as a

result it was not desired to limit the crawlers range of discovery. Currently the OCCS conducts short focused crawls, not continuous crawling, as a result the AdaptiveRevisitingFrontier is not applicable. However this may change and is discussed in more detail in section 7.4.2, which provides information on future work.

The Frontier in Heritrix is also responsible for performing URL canonicalisation tasks before the CrawlURI is passed to the Toe Thread in question. The Frontier checks the current URL against a list of already processed URLs. It is often the case that a URL for the same resource can be written in numerous ways. For instance:

- https://www.cs.tcd.ie/~slawless/index.html
- https://www.cs.tcd.ie/~slawless
- http://www.cs.tcd.ie/~slawless

All three of these URLs point to the same web page. Before comparing URLs the Frontier attempts to resolve this ambiguity by applying a list of canonicalisation rules to each URL. Some of the canonicalisation rules that can be applied are described below:

- BaseRule: Base of all rules applied canonicalising a URL.
- FixupQueryStr: Strips trailing question marks.
- LowercaseRule: Converts the entire URL to lowercase.
- RegexRule: General pre-defined conversion rule.
- StripExtraSlashes: Strips trailing slashes.
- StripSessionCFIDs: Strips cold fusion session ids.
- StripSessionIDs: Strips known session ids.
- StripUserinfoRule: Strips any 'userinfo' found on http/https URLs.
- StripWWWNRule: Strips any 'www[0-9]*' found on http/https URLs, if they have some path/query component (content after third slash).
- StripWWWRule: Strips any 'www' found on http/https URLs, if they have some path/query component (content after third slash).

CrawlOrder

The CrawlOrder is used by the CrawlController to assemble all the required components within Heritrix before initiating a crawl. All the configuration options for a crawl are also set in the CrawlOrder file. These options can be manually altered in advance of a crawl. The scope and frontier to be used are among the settings in the CrawlOrder.

The max-path-depth variable, which is set in the crawl order file, contains the maximum depth into a single site that the crawler will drill. There are two other variables in the crawl order file which add further limits upon crawl depth. max-link-hops and max-trans-hops specify the number of redirects which the crawler is allowed to follow for a single URI. These variables are used to prevent the crawler becoming stuck in malicious spider traps or in sites with dynamic content such as calendars.

The Heritrix CrawlOrder also contains three variables which can be used to impose limits on the duration and extent of each crawl. max-bytes-download can be used to terminate the crawl after a fixed number of bytes have been downloaded. If this variable is set to zero, the number of bytes is unlimited. max-document-download can be used to terminate the crawl after a fixed number of pages have been downloaded. Again, if the variable is set to zero, there are no limits placed upon page downloads. max-time-sec is used to terminate the crawl once a certain number of seconds have elapsed. This allows for very short crawls, or longer crawls on the scale of days if necessary. If one of these crawl limits is hit, a graceful termination of the crawl job will be triggered. However, any URIs already being processed will be completed.

Requirement vi of this research specified that the OCCS should respect the rights and desires of all content authors and site owners by obeying all robots.txt files. Heritrix provides five robots honouring policies which can be selected from and specified in the CrawlOrder file. The “classic” policy obeys all the rules specified in each robots.txt file encountered. The internet archive recommend the use of this policy unless special permission has been granted to crawl a site more aggressively. This honours all the wishes of the site owner and is the policy that is used for all crawls conducted by the OCCS. The other policies offered by Heritrix are: “ignore” which completely ignores all the rules specified by each robots.txt, “custom” which allows user defined rules rather than those discovered in each robots.txt file, “most-favoured” which obeys the rules set in each robots.txt file for the robot which is allowed most access or has the least restrictions and finally “most-favoured-set” is similar to “most-favoured” but with the addition of a set of robots whose rules the crawl can follow.

Processor Chain

The Processor Chain, illustrated in Figure 5-3 below, is a group of modular processors that perform specific tasks on each URI. These include fetching the URI, analysing the results of the fetch and passing newly discovered URIs to the Frontier for addition to the URI Queue. The processor chain is split into six linked sections and the crawler can be configured by adding to, removing from or re-ordering the processes contained within any of these sections.

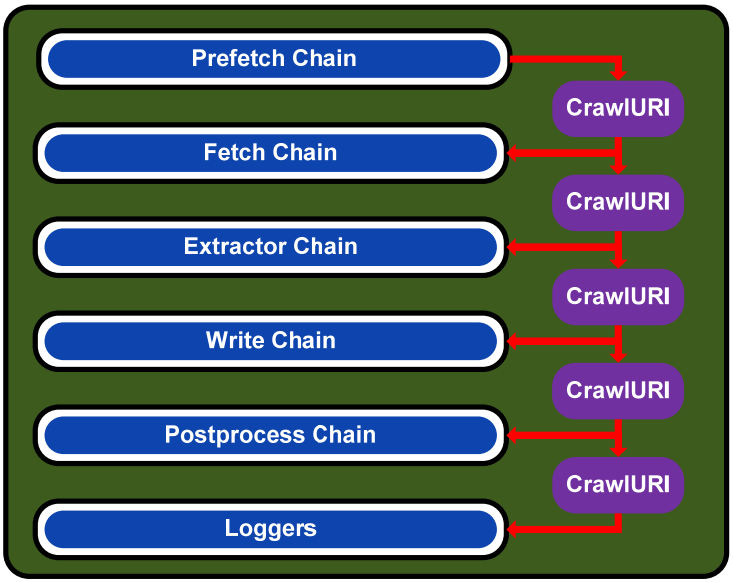


Figure 5-3 Heritrix Processor Chain

Each linked section contains zero or more individual processes. The order of these processes within each section is specified in the CrawlOrder. Particular processors only make sense within particular sections, for instance it would be illogical to have a HTML extraction process in the Postprocess chain. However this is not enforced so great care must be taken in the construction of the process chain.

The *Prefetch Chain* should contain the following two processors for all crawls.

- Preselector: This processor checks if the URI that has been passed in CrawlURI adheres to the scope of the crawl and should be included. This is a useful processor if the scope has been updated mid-crawl.
- PreconditionEnforcer: Ensures that all preconditions for crawling a URI have been met. This includes verifying that DNS and robots.txt information has been fetched for the URI.

The *Fetch Chain* retrieves the content for the current CrawlURI from the host server. There are two processors contained within this linked section, FetchDNS and FetchHTTP.

The *Extractor Chain* contains processors to search for URIs in the current CrawlURI. Heritrix has link extractors for all the most common web document types, HTML, CSS, JavaScript, PDF, MsWord and FLASH. It also has a "Universal Extractor" which attempts to extract links for any other format that is encountered.

Once a URI has been crawled and any new URIs of interest have been extracted by the Extractor Chain, the processors in the *Write Chain* store the crawl results. The data is written in the Internet Archive's ARC File format, which will be detailed at the end of this section. Extra processors can also be added to write in other data formats or index the newly crawled data.

The *Postprocess Chain* contains the following special purpose processors that should be included for all crawls regardless of scope.

- CrawlStateUpdater: Updates any information that may have been affected by fetching the current URI such as robots.txt and IP address information, this information is stored in the Crawler Server Cache.
- PostSelector: Checks all links extracted from the document at the current URI against the crawl scope. Those that are out of scope are discarded. It is possible to enforce the logging of all discarded URIs. PostSelector also schedules any URIs found in the current CrawlURI, that conform to the crawl scope, with the Frontier for crawling. Also schedules prerequisites if any exist.

Each crawl can be monitored and controlled from a Web Administration Console. This is a Jetty Java HTTP server application. Each crawl is passed a crawl order which is a list of instructions pertaining to the configuration of the crawler. The crawl order is also used to generate the scope of the crawl.

ARC Files

Heritrix stores the documents that it harvests in aggregate files of customisable size, which enables ease of storage in a conventional file system. This design attempts to ease the organisation and management of what could potentially be hundreds of millions of individual

documents. These aggregate files are structured according to the ARC file format. This format was designed to address several requirements, defined by the Internet Archive:

- The file must be self-contained. It must permit the aggregated objects to be identified and unpacked without the use of a companion index file.
- The format must be extensible to accommodate files retrieved via a variety of network protocols, including http, ftp, news, gopher, and mail.
- The file must be "streamable". It must be possible to concatenate multiple archive files in a data stream.
- Once written, a record must be viable. The integrity of the file must not depend on subsequent creation of an in-file index of the contents.

It is notable, however, that an external index of the contents of such archives and document-offsets greatly enhances the ability of search engines and similar systems to retrieve documents stored in the ARC format. ARC files are split into several discreet components:

- version-block – Identifies the original filename, file version, and URL record fields
  - filedesc – A special-case URL record
  - path – The original path name of the archive file
  - IP address – The address of the machine that created the archive file
  - date – The date the archive file was created
  - content-type – The format of the remainder of the version block
  - length – Specifies the size, in bytes, of the rest of the version block
  - version-number – Integer in ascii
  - reserved – String with no white space
  - origin-code – Name of gathering organization with no white space
  - URL-record-definition – The names of fields in URL records
- URL-record – Defines a document in the archive file. Provides the name and size of the object, as well as several other pieces of metadata, when available, about its retrieval.
  - url – Ascii URL string (e.g. "http://www.cs.tcd.ie:80/")
  - IP_Address – Dotted decimal identifier of host (e.g. 134.226.35.130 or 0.0.0.0)
  - archive-date – The date the document was archived
  - content-type – MIME type of data (e.g. "text/html")

- length – Ascii representation of size of document in bytes
- result-code – Http result code or response code (e.g. 200 or 302)
- checksum – Ascii representation of a checksum of the data.
- location – Url of re-direct
- offset – Offset in bytes from beginning of file to beginning of URL-record
- filename – Name of ARC file

- network_doc – The actual content of the document returned by the protocol
- doc – A combination of the network-doc and the URL-record

As noted, the most efficient way to retrieve a specific document from an ARC file is to maintain an index of document names, the ARC file in which they are located and the document offset within the ARC file. The retrieval of a document is then merely a matter of navigating to the ARC file in question and seeking the offset of the document.

5.2.5 Rainbow

Rainbow [Rainbow] is a statistical text classifier, developed by Andrew McCallum, and is integrated in the OCCS architecture to classify the content discovered by Heritrix and perform the “focussing” aspect of the crawl. Rainbow is based upon BOW [BOW], a collection of C libraries for text mining and retrieval produced at Carnegie Mellon University. It is implemented in the C programming language.

Rainbow analyses a content collection or a training collection and generates a model of the collection using term frequencies. Rainbow can then be deployed as a server, and using this model, can perform classification or diagnostics on any content which is passed to the server. In the case of the OCCS a model must be generated for an exemplar set of on-topic content and for a collection of off-topic content. Rainbow can then classify any content encountered in comparison to these two models.

When indexing content, Rainbow converts the content into a character stream, and subsequently converts this stream into tokens by a process called tokenisation or "lexing". Rainbow tokenises all alphabetic sequences of characters which are encountered in the file stream, any non-alphabetic characters are discarded. Each sequence of characters is then normalised. A stoplist of terms can be provided, and any term which appears on this stoplist

is discarded. There are several configuration options for the tokenisation process. Some of which can be useful when classifying web content, such as:

--use-stemming	This passes all words from the character stream through a Porter stemmer before calculating term frequencies.
--no-stoplist	This tells Rainbow to include words in the stoplist when generating the model. The default is to skip them.
--skip-html	This option is essential in the OCCS, it tells Rainbow to skip all characters between "<" and ">". Important when tokenising HTML files as it ignores all HTML structural tags.
--lex-white	This option tells Rainbow to simply grab space-delimited strings rather than tokenising the file with the default rules.

When the Rainbow model has been generated, content classification can be performed. Once deployed as a server, Rainbow reads the model from disk, then waits for a query document by listening on a network socket, the port of which is specified when deploying the server. When a document is passed to Rainbow, it is analysed, compared to the model and a relevency score is passed back.

Rainbow can perform document classification using one of several different strategies available in BOW, including Naïve Bayes, TFIDF with Rocchio weighting, K-nearest neighbor, Maximum Entropy, Support Vector Machines and Probabilitistic Indexing. The OCCS implements a Naive Bayes classification method using the multinomial, or unigram, event model. The multinomial event model specifies that content is represented by the set of term occurrences within that content. The order of terms within the content is lost and terms sequenced alphabetically. The number of occurrences of each term in the document is then captured. When calculating the relevance probability of a resource, the probability scores for each term that occurs are multiplied [McCallum & Nigam 98].

5.2.6 JTCL

TextCat is used in the OCCS architecture for language classification and filtering. The software is an implementation of the text categorisation algorithm presented in [Cavnar &

Trenkle 94]. Cavnar & Trenkle's technique is to calculate a "fingerprint" for each document encountered. A fingerprint is essentially a list of the most frequently occurring words in a document, ordered by frequency. This fingerprint is called an n-gram, which is defined as an "ordered sequence of words". When a fingerprint is calculated, it is compared with the fingerprints of a maintained corpus of documents. The documents from this corpus, whose fingerprints which most closely match the document currently being processed, are used to estimate unknown characteristics of the current document. Fingerprints are compared using a simple out-of-place metric.

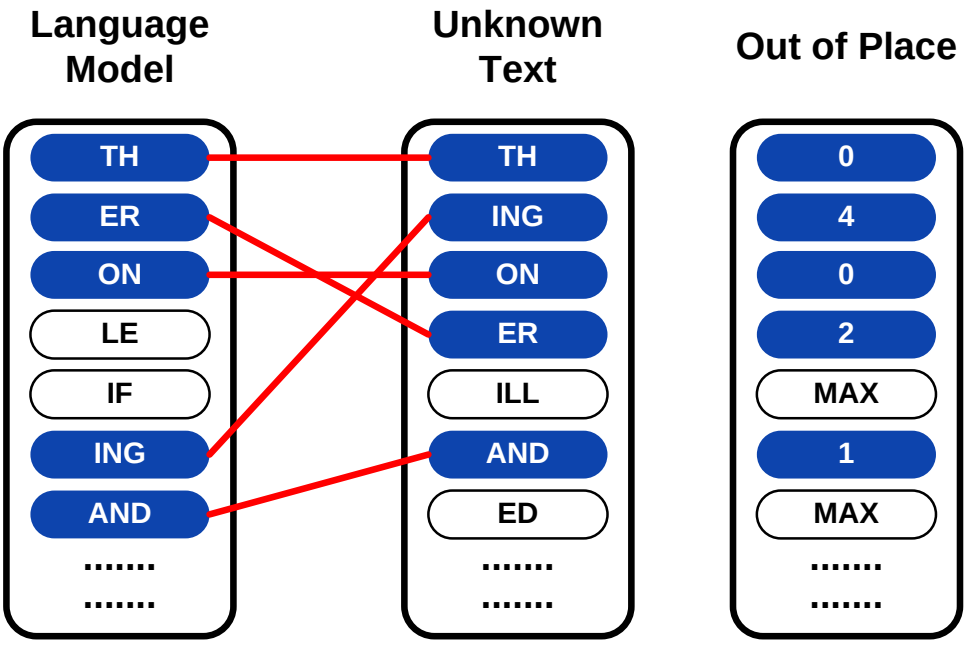


Figure 5-4 Out-of-Place Measure

The out-of-place measure, illustrated in Figure 5-4 above, corresponds to the distance that exists between the frequency rank of an n-gram in text being examined and its rank in an existing language model. This technique has been applied in TextCat to implement a written language identification program. All of these out-of-place scores are summed and the language of the unknown text under classification is deduced as being the language of the model that has the closest relative score. Currently, the TextCat guesser can identify 69 natural languages.

JCTL [JTCL], a Java implementation of the TextCat library, was chosen for inclusion in the OCCS architecture, as the Heritrix web crawler is implemented in Java. This implementation of the library was developed at Knallgrau New Media Solutions [Knallgrau]. The JCTL

implementation of TextCat can currently identify fifteen languages. For the purposes of this research we only needed to identify and harvest English language documents.

5.2.7 Lucene, Nutch and NutchWAX

Once a web crawl has been performed and a topic specific cache of content has been generated, it is necessary to index the cache so that it can be searched for specific information. Indexing is a method of analysing and classifying content to make it easier to retrieve. Details regarding each piece of content are saved in a data structure that is easily searched using keyword information.

NutchWAX [NutchWAX] is the indexing solution which is implemented in the OCCS, this is an extension of Nutch [Nutch] to include compatibility with web archive collections. Nutch itself extends the indexing methods of Lucene [Lucene]. To understand the indexing and query-document similarity functions of NutchWAX it is necessary to examine both Lucene and Nutch.

5.2.7.1 Lucene

In Lucene, the score of query *q* for document *d* correlates to the cosine-distance or dot-product between the document and query vectors in an IR VSM. A document whose vector is closer to the query vector is scored higher. The score for a particular document *d* for a query *q*, is the sum of the score for each term of the query. A terms score for a document is itself the sum of the term run against each field that comprises a document i.e. title, url etc. A document in a Lucene index is comprised of a set of fields. Per field, the score is the product of its term frequency, inverse document frequency, an index-time boost, a normalisation factor, a query normalisation factor, and finally, a factor with a weight for how many instances of the total amount of terms a particular document contains.

The score is computed as follows:

$$score(q,d) = coord(q,d) \times queryNorm(q) \times \sum_{t\ in\ q}(tf(t) \times idf(t)^2 \times t.getBoost() \times norm(t,d))$$

$tf(t)$ correlates to the *frequency* of term *t* in the currently scored document *d*. The default computation of $tf(t)$ in Lucene is:

$$tf(t) = frequency^{1/2}$$

$idf(t)$ is the Inverse Document Frequency of term *t*. The default computation for $idf(t)$ in Lucene is:

$$idf(t) = 1 + log\left(\frac{numDocs}{docFreq + 1}\right)$$

$coord(q,d)$ is a score factor based on how many of the query terms are found in the specified document. Typically, a document that contains more of the query's terms will receive a higher score than another document with fewer query terms. This factor is computed at search-time.

$queryNorm(q)$ is a normalising factor used to make scores between queries comparable. This factor does not affect document ranking since all ranked documents are multiplied by the same factor. Rather, it attempts to make scores from different queries or different indexes comparable. This factor is computed at search-time. The default computation for $queryNorm(q)$ in Lucene is:

$$queryNorm(q) = queryNorm(sumOfSquaredWeights) = \frac{1}{sumOfSquaredWeights^{1/2}}$$

The sum of squared weights of the query terms is computed by the query weight object. For example, a boolean query computes this value as:

$$sumOfSquaredWeights = q.getBoost()^2 \times \sum_{t\ in\ q} (idf(t) \times t.getBoost())^2$$

$t.getBoost()$ is a search-time boost of term *t* in the query *q* as specified in the query text, or as set by application calls to setBoost( ).

$norm(t,d)$ encapsulates three separate boost and length factors at index-time:

- Document boost is set by calling doc.setBoost() before adding a document to the index.

- Field boost is set by calling field.setBoost() before adding a field to a document.
- lengthNorm(field) is computed when the document is added to the index in accordance with the number of tokens of this field in the document. This is used so that shorter fields contribute more to the score.

When a document is added to the index, all the above factors are multiplied. If the document has multiple fields with the same name, all their boosts are multiplied together:

$$norm(t,d) = doc.getBoost() \times lengthNorm(field) \times \prod_{field\ f\ in\ d\ named\ t} f.getBoost()$$

The resulting *norm* value is encoded as a single byte before being stored. At search-time, the norm byte value is read from the index directory and decoded back to a float *norm* value. This encoding/decoding process, while having the benefit of reducing index size, comes at the cost of precision. It is not guaranteed that $decode(encode(x)) = x$. For example, $decode(encode(0.89)) = 0.75$.

5.2.7.2 Nutch

Nutch is built on Lucene and implements Lucene's scoring mechanism with a few small alterations. Nutch generates a total score for a page, a score per query term and a score per query document field. The score for a particular field comprises a query component and a field component. The query component includes a query-time boost, an idf that is the same for the query and field components and a queryNorm.

Scoring in Nutch can be influenced by changing the query-time boosts. The default query-time boost values in Nutch are:

query.url.boost	4.0f
query.anchor.boost	2.0f
query.title.boost	1.5f
query.host.boost	2.0f
query.phrase.boost	1.0f

Query terms found in within a document URL get the highest boost, followed by terms found in anchor text. Anchor text makes a large contribution to page score.

5.2.7.3 NutchWAX

NutchWAX stands for Nutch with Web Archive eXtensions. The web archive extensions include an adaptation to the Nutch Fetcher to retrieve content from web archive collections stored in the Internet Archive's ARC file format. Plugins have also been added to Nutch to add extra, ARC-specific, fields to the index, such as arcoffset and arcfilename.

NutchWAX stores numerous metadata fields containing details about each content object within the ARC. The full list of fields is as follows:

- arcoffset – The offset into the ARC file at which the content begins.
- primaryType – Content Type e.g. text
- subType – Media Type e.g. html
- type – Combination of primaryType and subType e.g. text/html
- title – The title of the content
- arcdate – Date of content archival
- content – Full text of the content
- date – Date of indexing
- boost – A decimal value that effects page ranking based upon where a query term occurs in an archived document. Boost values can be altered but by default they are:
    - url.boost, 4.0f
    - anchor.boost, 2.0f
    - title.boost, 1.5f
    - host.boost, 2.0f
    - phrase.boost, 1.0f
- arcname – Name on the ARC file in which the content is contained
- segment – Segment of the NutchWAX index in which the ARC file is located
- contentLength – Length of the content
- encoding – Form of text encoding used in the content
- host – Section of the content URL relating to Host
- exacturl – Query for an explicit URL

- url – URL from which the content was harvested
- digest – Used for HTTP authentication
- collection – The collection to which the ARC file belongs
- anchor – Anchor text relating to the site link

NutchWAX uses the Hadoop [Hadoop] framework, upon which it runs its indexing jobs. Hadoop is an open source implementation of Google's mapreduce [Mapreduce] and GFS utilities [GFS]. Mapreduce is a distributed computing paradigm where an application's workload is divided into many small fragments of work, each of which may be executed or re-executed on any node in a cluster of computers. GFS is a distributed file system that stores data on these nodes, providing very high aggregate bandwidth across the computing cluster. Both of these utilities are designed so that node failures are automatically handled by the framework.

A list of the ARC files that comprise the content cache are provided to NutchWAX which imports these in turn for indexing. The ARC files are sequentially imported, all the content objects or URLs are extracted, parsed and indexed. This process is repeated until all the ARC files in the cache have been processed. Upon completion an index is created for the entire collection of ARCs. NutchWAX can be deployed under a servlet such as Tomcat to provide a free-text search interface for the content cache. This allows the input of a search string which is used to perform a search across the entire content cache for candidate content objects. The results are ranked according to the Nutch relevancy scoring mechanism detailed above.

5.2.8 WERA

Two systems which enable the linking of a content index with a web archive and visualisation of the archived material were researched. The Internet Archive's Wayback Machine [Wayback] and the open source project WERA [WERA]. At the time of development, the Wayback machine was not compatible with NutchWAX so WERA was selected to implement the visualisation of search results in the OCCS.

WERA is an acronym of "Web Archive Access". It is an archive viewer application that gives access to ARC file collections as well as the ability to perform free-text search. It also enables users to navigate between different historical versions of a web page, this occurs as

some web crawlers recursively crawl particular sites to capture additions and updates. WERA is based upon the NwaToolset [NWA]. It is implemented in PHP and Java. WERA is used in combination with an index of a web archive, such as a NutchWAX index, the collection of ARC files and a interface to the web archive, called ARCRetriever, from the NWA toolset. The user interacts with the WERA interface and submits a keyword-based query. WERA uses this query to generate a search request which is sent to NutchWAX. The important query parameters that NutchWAX expects are:

query	Contains the keyword-based query submitted to the interface
hitsPerPage	Specifies how many results should be listed on each page.
hitsPerDup	Specifies how many of dedupField can be returned in the results field. If you were de-duplicating by URL you may want to allow more than one result from a single URL to appear in the list.
dedupField	Specifies which field to run de-duplication on. This ensures that duplicate results are not listed.

Based upon this search query NutchWax conducts a query-document similarity rating using the Nutch scoring mechanism and constructs an a9 OpenSearch RSS formatted result set, which is an XML formatted file, and sends this back to WERA. The result set is formatted by WERA and output to the user on the interface. When the user clicks on a result in the ranked list, WERA constructs a request to ARCRetriever. Each request contains three parameters:

reqtype	In the case of the OCCS this is always “getfile” which informs ARCRetriever to retrieve the actual content resource
aid	This is a combination of the name of the ARC file where the content resides and the offset within that ARC file where the

content can be found.

Both the ARC name and offset fields are stored in the NutchWAX index. WERA receives an archived content resource from ARCRetriever. A javascript link rewriter is inserted in the resource to ensure that links within the content point to WERA rather than out to the WWW. Before WERA displays the resource in the browser, header information on content type and encoding is set according to values which are stored in the NutchWAX index and were provided in the result set. This ensures that the content renders correctly in the browser.

5.2.9 Contribution of the Author

It was neither practical nor logical to re-invent tools to implement the discovery, classification, harvesting and delivery of content from the WWW. Instead the author has implemented, enhanced and integrated various open source solutions to create a novel architecture for the leveraging of open corpus content for use in TEL. The open source systems utilised in the development of the OCCS are described in the above sections. These tools are Heritrix, Rainbow, JTCL, NutchWAX and WERA. The author's contribution to the development of this architecture is:

- The seamless integration of these disparate systems to develop a tool chain which creates topic specific caches of educational content for use in TEL.
- The authoring of Perl scripts to train and fine tune the Rainbow text classifier, and integrate it into the Heritrix process chain.
- The integration of JTCL into the Heritrix process chain to filter content by language before being processed by Rainbow.
- Integration of NutchWAX and WERA which enabled WERA to use the Index created by NutchWAX to locate appropriate content and use the ARCRetriever to extract the desired content from the Heritrix cache.
- Re-authoring of WERA interface to create the OCCS search interface.

5.2.10 OCCS Component Integration

The previous sections described the individual open source components that were implemented as part of the OCCS process flow to satisfy the design requirements of the service. The seamless integration of each of these components was a complex task, which was necessary to ensure the correct operation of the OCCS as a complete tool chain. The integration of these components is described in the section below.

5.2.10.1 Heritrix, JTCL and Rainbow Integration

To integrate both JTCL and Rainbow into the Heritrix web crawler, it was necessary to author new processes which could be plugged into the Heritrix processor chain. These new java processes were added to the Heritrix extractor chain, as illustrated in Figure 5-5 below, which is called directly after the text of each URI has been fetched.

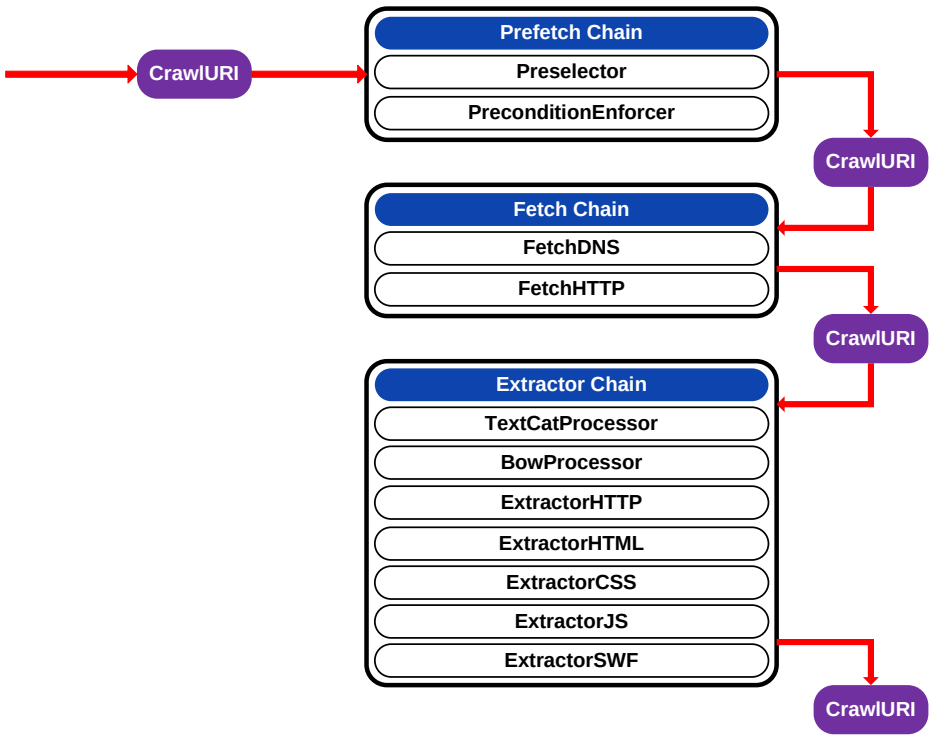


Figure 5-5 Rainbow and JTCL Integration

JTCL was added as the first processor in the extractor chain. A new Java module called LanguageModule and class called TextCatProcessor were added. This Class passes the CrawlURI to JTCL. The content which resides at the CrawlURI is fetched and a parser is used to extract the text from the document by stripping all the formatting tags. The parser used is dependent upon the format of the resource. JTCL analyses this content stream and compares it to stored language modules. JTCL passes back a language estimation to the extractor chain. If the estimation is “English”, the CrawlURI is passed to the next processor in the extractor chain, which in the OCCS is BowProcessor.

BowProcessor is a Java Class that is added to the extractor chain directly after TextCatProcessor. This class in turn calls RainbowClassifier. The CrawlURI is passed to BowProcessor for any content deemed by JTCL to be English. Various manually specified

parameters from the configuration file are loaded, such as CutOff and Rainbow Port. The content of the CrawlURI is fetched and parsed according to its format. All structural tags are stripped and the text stream is extracted. This text stream is then passed to the RainbowClassifier class and classification results are returned. This result is compared to the CutOff relevance boundary. If the content achieves a rating above CutOff then the CrawlURI is passed to the next processor in the extractor chain. The next processors extract any URIs contained within the text and add them to the URI Queue in the Frontier. If the content achieves a rating below CutOff, the remainder of the extractor chain is skipped. Each URL and its respective classification score are recorded in a log file.

RainbowClassifier is a general purpose Java class for communicating with the Rainbow server. A socket connection is established with the Rainbow server. The server must be running on the host and port specified in the configuration file which was provided to Heritrix. The server is queried by the classify method. This method requires an input text string, and returns classification results in the form of a class (or category) followed by scores for all the classes. In the OCCS there are two classes: positive (on-topic) and negative.

These new Java Classes are included in the processor chain by the addition of a reference in the CrawlOrder or its XML representation order.xml [Appendix C].

5.2.10.2 WERA and NutchWAX Integration

NutchWAX offers a keyword-based search interface which could have been used to visualise query results for the OCCS. However, there was a problem with the visualisation of the content of search results from the NutchWAX ranked list. The majority of search engines create a cache of content as they crawl the web, essentially this cache is comprised of a copy of every web site encountered by the web crawler. In the case of our crawler, this cache is comprised of copies of all web sites that pass the classification step of the crawl. This cache of content is then analysed to create a searchable index for query resolution and content identification. However it is at this point that the requirements of this research differ from the majority of search engines. When a search is performed on a regular search engine and a result clicked on, the user is redirected back to the original URL where the content was discovered. This is how visualisation occurs in the NutchWAX interface. However, in the case of this research, it is necessary to redirect the user to the actual content contained within the web archive. This is necessary as further manipulation of the content cache can be

executed after the content has been harvested. Future research and development in the areas of metadata enhancement, metadata mappings, content slicing and automatic annotation can add valuable semantic tools to the OCCS process chain and could tailor the content being delivered to the TEL environment.

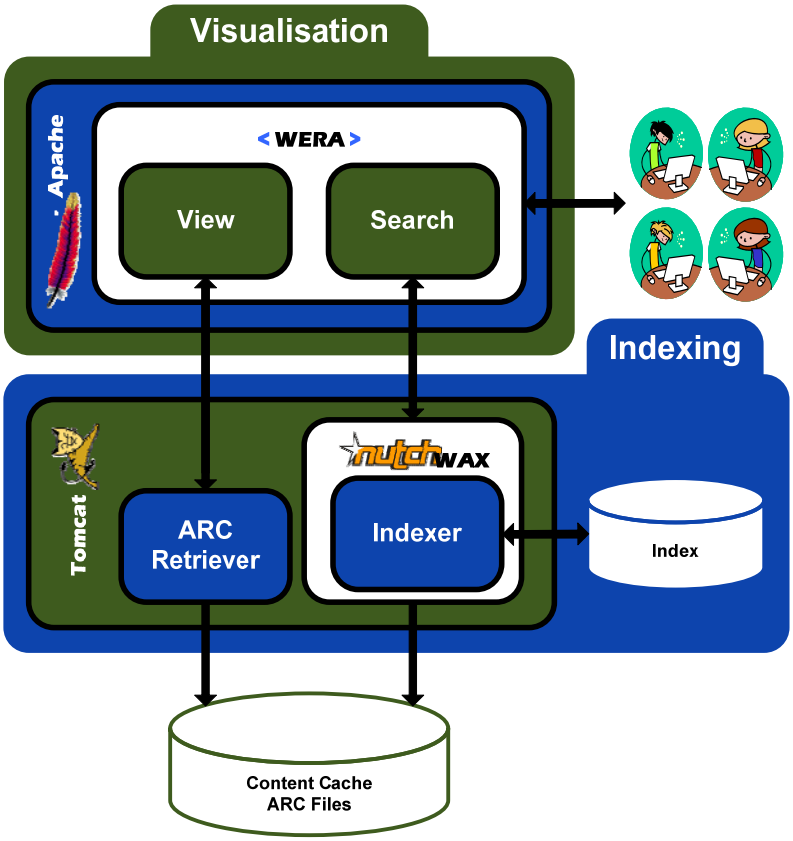


Figure 5-6 Visualisation Architecture

WERA provides a visualisation tool which solved this problem, allowing the extraction and display of content from within the web archive, as illustrated in Figure 5-6 above. An interface is provided which supports searching, browsing and navigating the archived web pages. When the user submits a query, WERA uses the NutchWAX index to find relevant archived files that satisfy the query.

Figure 5-7 OCCS Wera Search Interface

The Nutch OpenSearch XML result set is transformed into a PHP array for WERA compatibility. The user can also search for a specific URL, at which point WERA will return the archived file originally downloaded from that URL. A web application called ARCRetriever is used to extract the archived version of the page from the web archive rather than redirect the browser to the original WWW URL. WERA also possesses the ability to display the metadata fields pertaining to the returned content.

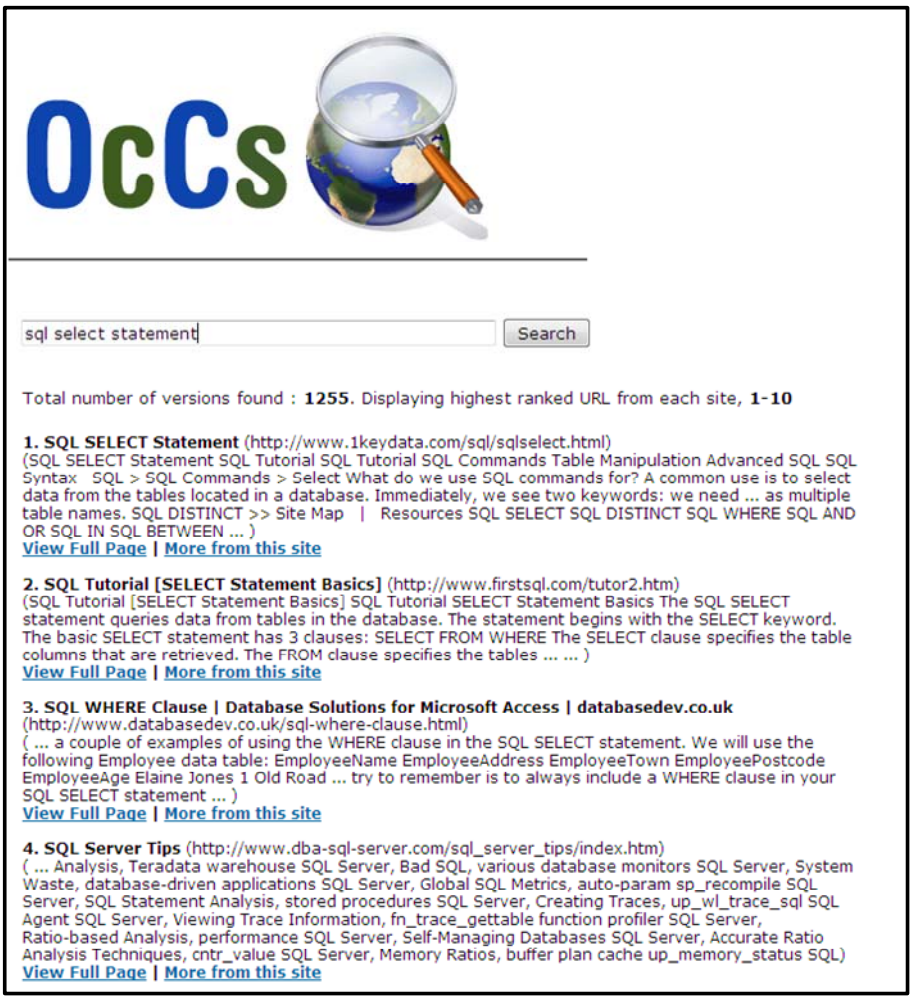


Figure 5-8 OCCS Wera Search Interface

WERA provides timeline functionality in its user interface for the browsing of single resources that have been recursively crawled over a period of time. However it was decided that the OCCS search interface should be kept as clean and simple as possible. As detailed in section 4.2.2, many evaluations have found that users prefer simplistic user interfaces when searching for content. Figures 5-7 and 5-8 above, demonstrate the OCCS implementation of the Wera Search Interface. The usability of the interface was very important and the timeline functionality of WERA was deemed unnecessary for the tasks which will be performed using the OCCS.

5.2.11 Crawl Preparation

Before Rainbow can be used for classification, it must be trained for the subject area in question. This involves indexing a set of representative documents for the subject area in question and creating a statistical model of the subject area. The set of training documents is

provided to the Rainbow classifier. The documents are placed in a directory or directories, one per category. In the case of the OCCS there are always two categories: positive, or on-topic; and negative, or off-topic. Rainbow indexes the supplied documents and uses them to build a statistical model of the corpus, which is stored on disk and used by the classifier during a web crawl.

Two external web services are used during the generation of a Rainbow training set. These include the Open Directory Project (ODP) and the Google API. Before detailing the process of training Rainbow for a specific domain, the author will provide a brief introduction to these services.

5.2.11.1 Open Directory Project

The ODP [ODP] is arguably the largest, most comprehensive and most widely distributed, human-edited directory of the WWW currently available. The ODP is often referred to as DMOZ, an acronym for directory.mozilla.org, and is hosted at dmoz.org. The ODP forms a hierarchical topic structure of the WWW. The ODP essentially consists of a human-edited directory of the web containing upwards of 6 million websites, classified into over 600,000 categories. The hierarchy has seventeen top-level categories which divide the WWW by topic, these categories are:

- Adult
- Arts
- Business
- Computers
- Games
- Health
- Home
- Kids and Teens
- News
- Recreation
- Reference
- Regional
- Science
- Shopping

- Society
- Sports
- World

The ODP is maintained by a group of more than 70,000 volunteer editors, who assess every website submitted for validity and relevance before categorising the site and adding it to the directory. ODP guidelines [ODP Submission] must be strictly adhered to and the site must be directly relevant to the category chosen, or the submission will be rejected. ODP categories therefore provide a volume of websites that can be assumed, with a high degree of confidence, to be accurately on-topic. ODP data is provided for open use in Resource Description Framework XML (RDF) format. The entire ODP database can be downloaded for use as an RDF dump. Google and many other web services incorporate the ODP database to aid with the categorisation of content.

5.2.11.2 Google API

Google offers an API [Google API] to perform freetext searches and retrieve results lists much as you can through any of the google search sites. The API takes input, in the form of search terms, and returns a list of *n* results, ranked according to Google's interpretation of relevance.

Google uses the SOAP protocol [SOAP] and WSDL [WSDL] for sending data between the API server and external applications. The OCCS uses Net::Google [Net::Google], a package of modules from CPAN [CPAN], developed by Aaron Straup Cope, that provides a simple interface in Perl to the Google Search API. A number of parameters can be passed to a Net::Google search instance to form a Google API call.

- *key:* Sets the Google API License Key.
- *query:* The keyword string for which the search is to be performed.
- *http_proxy:* Sets the HTTP proxy if required.
- *max_results:* The default number of results returned by and API call is 10. However, if a number greater than 10 is added to a search, the *results* method will make multiple calls to Google API.
- *restrict:* Allows restrictions to be placed on a search, for example, restrict(qw(countryIE)) would return only results for Ireland.

- *filter:* A boolean parameter that states whether the results should be filtered.
- *safe:* A boolean parameter that states whether safe-mode should be on.
- *lr:* Allows restrictions on the language of the search, for example lr(qw(en)) will return English language results only.
- *response:* Returns an array of *Net::Google::Response* objects, from which the search response metadata as well as the search results may be obtained.
- *results:* Returns an array of *Result* objects, each of which represents one result from the search.
- *queries_exhausted:* Returns true or false depending on whether or not the current in-memory session has exhausted the Google API 1000 query limit.

The results list is returned as an array which can easily be looped through and processed as desired. Search results include URL, document title, ODP category to which the document belongs, ODP summary for the document if one exists, and the document size. Currently Google restricts the number of searches a user can make to 1000 per day.

5.2.11.3 Training and Seeding Mechanism Execution

The Rainbow training process, illustrated in Figure 5-9, is implemented in the OCCS as a series of scripts that need to be configured and executed before a focused crawl on a subject domain can take place. The scripts and the files they generate set the scope and focus of the crawl by training the classifier to recognise content relevant to the subject of interest. The training scripts also set a relevance quotient that dictates acceptance or rejection of a URL during the crawl. Three initial input files need to be created to begin the execution of the training process.

The configuration file is used throughout the training and crawl setup process. It contains a number of variables that affect the performance of both Rainbow and Heritrix. The fields contained in the file are as follows:

- The path to where output files should be written
- The path to Rainbow
- The path to Heritrix
- The topic of the crawl
- The max number of URLs to crawl

- The max number of results from Google API
- The number of stopwords that Rainbow should generate
- The number of negative categories to generate
- An indicator to identify if keywords are provided
- An indicator to identify if ODP Categories are provided
- An indicator to identify if there are OAI repositories provided
- The classification relevancy boundary
- The license key for use with the Google API
- The IP address of the Rainbow host machine
- The Proxy for Internet Traffic
- The port on which the Rainbow server is to run
- The username for the proxy, if necessary
- The password for the proxy, if necessary

The keywords file contains a list of manually collated keywords that directly reflect the subject of interest to the crawl. These keywords are used in combination with the Google API to generate the training set for Rainbow so they should be highly specific and focused on the subject area.

The ODP Categories file contains a list of ODP categories which accurately encompass the topic of the crawl. Content is extracted from these categories and added to the positive training set of examples for training Rainbow.

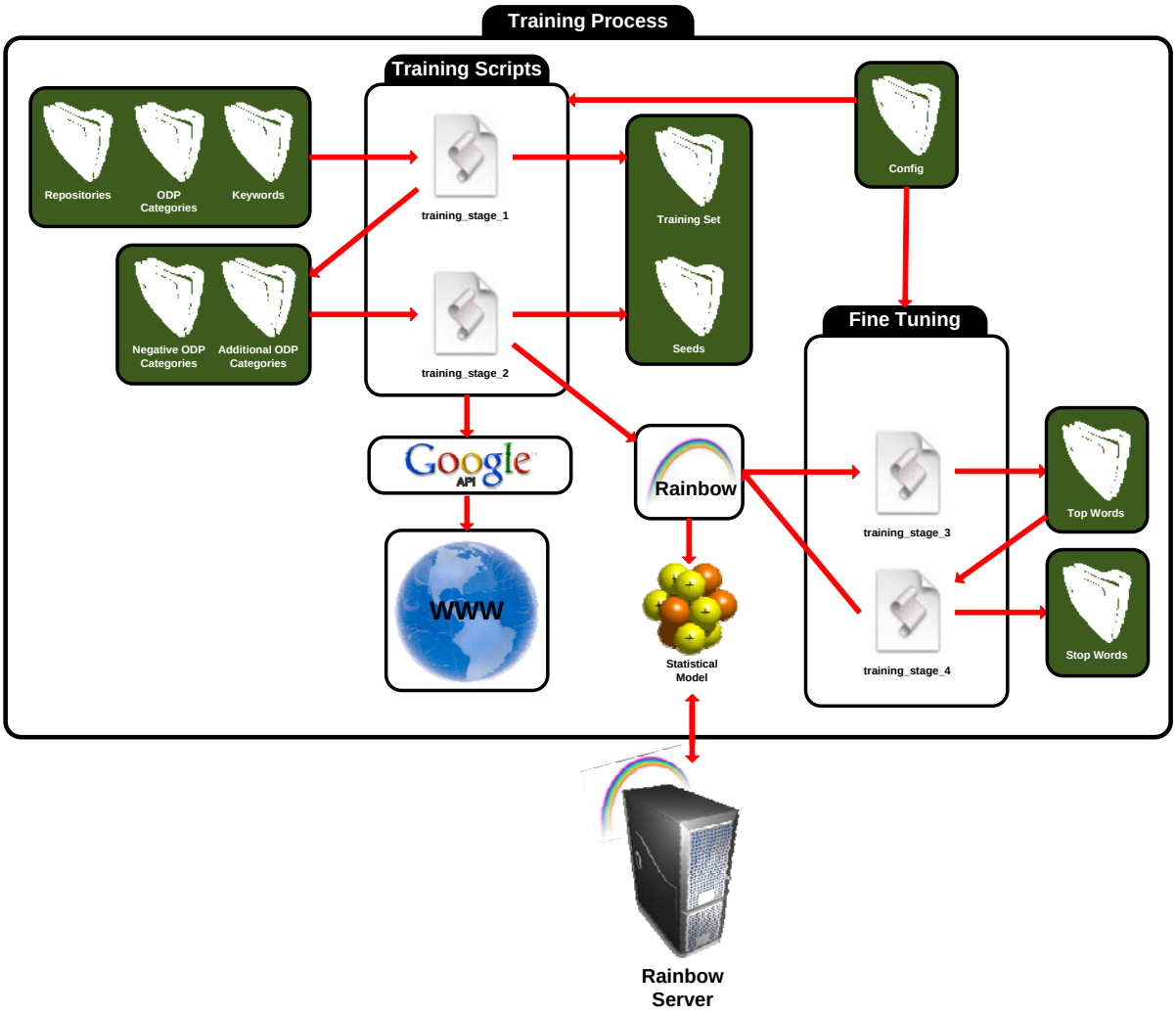


Figure 5-9 Rainbow Training Process

The scripts used to train the Rainbow classifier and produce the statistical model of the subject area need to be executed sequentially in the following fashion. Figures 5-10 to 5-14 illustrate the inputs and outputs for each script.

Inputs	Script	Outputs
Configuration Info Keywords ODP Categories	training_stage_1	Additional ODP Categories Negative ODP Categories Seeds Training Set

Figure 5-10 Rainbow Training Script 1 – Inputs and Outputs

For each keyword in the keywords input file, the script forms a search query and sends it to the Google API. A set number of results are returned, specified by the max-search-results

variable in config.txt. The content of the results are added to a directory as positive examples for the classifier training set. The top listed URL is also added to the seeds.txt file as a seed URL for the crawl.

The second part of the Rainbow training set generation involves the positive ODP categories supplied. An ODP category is provided as an input parameter to a script. This script queries a MySQL database, which has been populated using an RDF dump of the ODP category structure and content. This produces content files belonging to that category. This set of positive content examples is extracted for each ODP category provided in the input file.

Describing the negative class in a classification task is often problematic. This is not just a visualisation problem, but can also significantly affect the accuracy of learning algorithms [Chakrabarti et al. 99]. As a result, a thorough negative class generation process was required in the OCCS. A second function takes the list of ODP categories provided as input. A function is performed on the MySQL database which generates the set difference of the universal ODP category set and the input set. This list of categories is then returned. The list of negative ODP categories returned is then used to generate the negative training set for the classifier. The content contained in each category is extracted and added to the negative training set.

The URLs obtained from the ODP database, in both the positive and negative generation steps, are fed to a HTML-to-text parser to extract the content. These text documents are then added to the corresponding training set. Any ODP categories encountered in the results list of the Google API calls that are not listed in the ODP Categories input file are saved as additional ODP categories. Once this training step has finished, the additional ODP categories file needs to be manually edited to tag the valid entries.

Inputs	Script	Outputs
Configuration Info Additional ODP Categories Negative ODP Categories Training Set	training_stage_2	Training Set Classification Model StateInfo Rainbow Process

Figure 5-11 Rainbow Training Script 2 – Inputs and Outputs

The process of extracting the content contained within the ODP categories is repeated for any ODP categories that have been tagged as valid in the additional ODP categories file. Once this is complete, the rainbow classifier is trained. This is accomplished by passing the full training set, both positive and negative, to the classifier for classification. This generates the statistical model of the training corpus using the methods described in section 5.2.4 above.

After Rainbow is trained, the classifier is launched as a continuous background query server process, or daemon, on the port specified in the config file. This facilitates a simple interface between Rainbow and Heritrix during a focused crawl. The process id of this Rainbow query server is then added to the StateInfo file.

Inputs	Script	Outputs
Configuration Info Rainbow	training_stage_3	Top Words

Figure 5-12 Rainbow Training Script 3 – Inputs and Outputs

To improve the accuracy of the statistical model produced by Rainbow certain diagnostic routines can be used to fine-tune the classification. The diagnostic option print-word-probabilities is used in this script to generate a list of the most frequently occurring terms in the positive training set. This set of keywords can be seen to describe the subject area. The number of words added to the list is specified in the config file by the variable topN. These keywords are then used to generate the top words file. This file needs to be manually edited to tag all terms which are invalid and should be added to the stopword list. Some invalid terms may occur frequently in the positive training set even when they are unrelated or may be too general to be of value in content classification.

Inputs	Script	Outputs
Configuration Info Top Words Rainbow	training_stage_4	Classification Model Stop-List Rainbow StateInfo

Figure 5-13 Rainbow Training Script 4 – Inputs and Outputs

This script uses the tagged top words file to retrain the rainbow classifier. This is achieved by using the append-stoplist-file option within the classifier. Any terms in the top words file that have been tagged as erroneous are added to the Stop-List file. While the use of training stages three and four is optional, it is highly recommended, as it can provide a noticeable improvement in content classification. Multiple iterations of these scripts should be executed until no erroneous terms are appearing in the top words file. The statistical model is then regenerated. Once this is complete the classifier process is killed and the classifier restarted to utilize the newly generated model. The StateInfo file is then updated with the new process id for the Rainbow Server.

Inputs	Script	Outputs
Seeds Configuration Info Rainbow order.xml	start_heritrix	Heritrix StateInfo

Figure 5-14 Heritrix Initiation Script

This script starts the Heritrix crawler and initiates a crawl. Several parameters are passed to Heritrix in the startup command:

- The location of the Rainbow and JTCL configuration files
- The maximum allowed java heap size
- The path to the Heritrix startup script
- The path to the order.xml config file

order.xml is an XML representation of the CrawlOrder, an example can be found in Appendix C. This file contains configuration variables for all the following crawler functions:

- scope
- frontier
- http-headings
- robots-honouring-policy
- uri-canonicalisation-rules
- pre-fetch-processors

- fetch-processors
- extract-processors
- write-processors
- post-processors
- loggers
- recovery-path

Once the crawler starts, the Heritrix process id is added to the StateInfo file. As Heritrix traverses the WWW, JTCL is used for language identification and Rainbow is used to compare all English language content encountered to the statistical model of the subject area and rate it for relevancy. A relevancy boundary is manually set in advance of each crawl and this dictates how accurate to topic the content must be judged, to be deemed relevant and included in the cache. If the content is adjudged to be accurate enough to the subject area to achieve a rating above this boundary then the content is passed along the Heritrix extractor chain for URI extraction. This means that this web document is added to the content cache and any links discovered within the document are added to the URI Queue in the Frontier. If the content achieves a rating below the relevancy boundary it is discarded and no URI extraction is performed.

5.2.12 Summary

The design of a content discovery and retrieval tool, the OCCS, was described in chapter 4 of this thesis. A series of technical requirements were defined, which the OCCS must implement to satisfy the objectives of this thesis. This section described the technical implementation of the OCCS which satisfies these requirements. The individual components of the OCCS were described, and how they relate to the technical requirements. The open source tools which were used to implement these components were then described in detail. The integration of these individual solutions to form a novel, seamless tool-chain was then detailed.

5.3 The User-driven Content Retrieval Exploration and Assembly Toolkit for TEL

In TEL applications based upon pedagogically-informed design, learners become responsible for their educational progress and are thus more actively engaged. U-CREATe and OCCS combine to create one such system which supports educational theory through a novel, practical solution. U-CREATe is a TEL application, designed to allow learners to create

Mind maps which detail the topic which they are exploring. More detail about Mind mapping can be found in Appendix A. As a topic is explored and a Mind map created, content specifically related to the topic in question is supplied by the OCCS to aid the learner. This application serves to demonstrate and validate the use of open corpus content in TEL scenarios.

The design of a TEL application which enables the incorporation and resource-level reuse of content from a cache generated by the OCCS during the execution of a pedagogically meaningful educational experience was described in chapter 4 of this thesis. A series of educational requirements were defined. Based upon these, a set of technical requirements were defined which U-CREATe must implement to satisfy the objectives of this thesis. These technical requirements are:

The technical requirements of U-CREATe can be summarised as follows:

i. Implement a Graphical User Interface which supports the generation of Mind maps.
ii. Utilise and extend existing open source solutions during the GUI implementation.
iii. Caches of complementary contents should be provided to support the learner during the authoring of a Mind map.
iv. These content caches should be provided by the OCCS and the OCCS search functionality should be accessible via the U-CREATe GUI.
v. The learner should be able to browse and explore selected content in a separate window without navigating away from their Mind map.
vi. A learner should be able to tag each node in a Mind map with hyperlinks to specific pieces of content from the OCCS cache.

A set of usability guidelines were also defined in section 4.4.3, and these guided all aspects of the design and implementation of the U-CREATe interface. These guidelines can be summarised as follows:

- The interface should be consistent in all aspects. It should use standardised terminology, colours, fonts etc.
- Shortcuts should be provided for experienced users through function keys and hidden commands.

- Frequent or minor user actions can produce subtle responses while infrequent or major user actions should produce more significant responses.
- Dialog boxes should be used to signify the completion of action sequences.
- Design a system to prevent errors, but if they do occur the system should detect them and offer simple feedback for handling the error.
- Provide functionality so that users can easily undo or redo actions automatically.
- The user should always be the initiator of an action rather than a respondent.
- Reduce Short-Term Memory Load by providing action sequences with no more than seven plus or minus two actions.

Three open source Mind mapping solutions were reviewed in Appendix A in line with requirements **i** and **ii**. Freemind [Freemind] is a graphical mind mapping application integrated with an easy to operate hierarchical editor. Hyperlinks to the WWW or local file system can also be added to nodes which could be configured to satisfy requirement **vi**. Freemind is implemented in Java and available under the GNU GPL [GPL]. It can be installed on the majority of operating systems including Microsoft Windows, various Linux distributions and Apple OS X.

Labyrinth [Labyrinth] is a lightweight mind mapping tool. The project page states that the system is intended to be as light and intuitive as possible, while still providing a wide range of features. Labyrinth is implemented in Python and uses the GTK graphics toolkit and Cairo library for it graphics rendering. However, Labyrinth is intended for use as part of the GNOME [GNOME] desktop project on Linux and Unix operating systems which limits its use within the scope of this research as the vast majority of desktop machines within Trinity College student labs run Microsoft Windows. The Visual Understanding Environment (VUE) [VUE] combines presentation software with a concept and content mapping application. VUE is quite feature rich and provides additional functionality for the creation of ontologies defined using RDF-S or OWL. VUE is implemented in Java and can be installed on the majority of operating systems.

It was decided that Freemind was the most applicable tool for use in this research. It is actively developed and has a lively developer community. It is also a stand-alone mind mapping editor and not part of a larger suite of integrated tools. It also has windows

functionality and features which satisfy the majority of the technical requirements of U-CREATe.

5.3.1 Contribution of the Author

As described in previous sections, it was neither practical nor logical to re-invent tools to implement the implement the mind map functionality required by U-CREATe. Instead the author has enhanced the open source software, Freemind. The author's contribution to the development of this application is:

- Enhancement of the functionality of Freemind to integrate OCCS access into the interface to allow content exploration during Mind map authoring.
- Complete re-authoring of Freemind's capabilities regarding the addition of content hyperlinks to a Mind map to allow:
    - The addition of multiple hyperlinks per node
    - The easy browsing of all hyperlinks associated with a node
    - The easy deletion of hyperlinks associated with a node

5.3.2 Freemind

Freemind [Freemind] is an open source, graphical Mind mapping tool. Freemind is licensed under the GNU General Public License (GPL) [GPL]. This means Freemind is available to use for any desired purpose without commanding a license fee. It also means that any code developed upon, or derived from current Freemind code must also be licensed under GNU-GPL. As a result, U-CREATe, when complete and stable, will be made available under the GNU-GPL. Freemind is implemented in Java using Swing [Swing], a widget toolkit for the development of graphical user interfaces (GUI). Freemind provides Mind mapping functionality integrated with an easy to operate hierarchical editor.

Freemind implements a multi-tier architecture based upon the Model-View-Controller (MVC) [MVC] paradigm. Complex computer applications which process large amounts of data and present large volumes of information to the user, often separate data models from user interface concerns. As a result, changes to the user interface do not affect how the application handles information, and vice versa, reorganisation of application data does not require changes to the user interface. The Model-View-Controller architectural design is one method of implementing the decoupling of content and system logic from information presentation and user interaction. This is achieved through the introduction of an intermediate

system component; Controller. The MVC pattern of architectural design was first developed by Trygve Reenskaug, then working at Xerox PARC [Reenskaug 79].

Systems are commonly split into three component layers; presentation, system logic and data access. In MVC the information presentation layer of the system is further divided into two new components, View and Controller, while the data access and system logic layers are grouped together in the Model component.

- **Model:** Contains the domain-specific representation of objects contained within the system. The model also defines the system logic for accessing and altering these defined objects. Model objects are not directly displayed but are rendered by the View component. The MVC design enables models to be exported and reused by other systems.
- **View:** Renders the model into a user interface for display and interaction. Multiple views can exist for a single model for different purposes.
- **Controller:** Acts as an intermediary between the Model and View components. The Controller also listens for, processes and responds to user interface events, such as mouse movements, keyboard strokes or interface actions. These events can require changes to the model. The controller communicates these changes to the Model component. Typically there is one Controller per window. In many applications the Controller is tightly coupled to the View.

MVC applications can take varying forms and follow many marginally different control flows, however typically MVC systems and in the case of this research, Freemind, function in the following manner:

- A user interacts with the user interface by moving the mouse, pressing a keyboard "hot key" or executing an interface action.
- Event listeners in the Controller are alerted by this user-triggered input event
- Depending on defined actions based on the user interface event in question, the Controller may access the model and update it as required.
- View uses the model to generate the user interface display. View can request the state of the model and be notified of any updates which require the regeneration of the display.
- The user interface awaits further user interaction, completing the cycle.

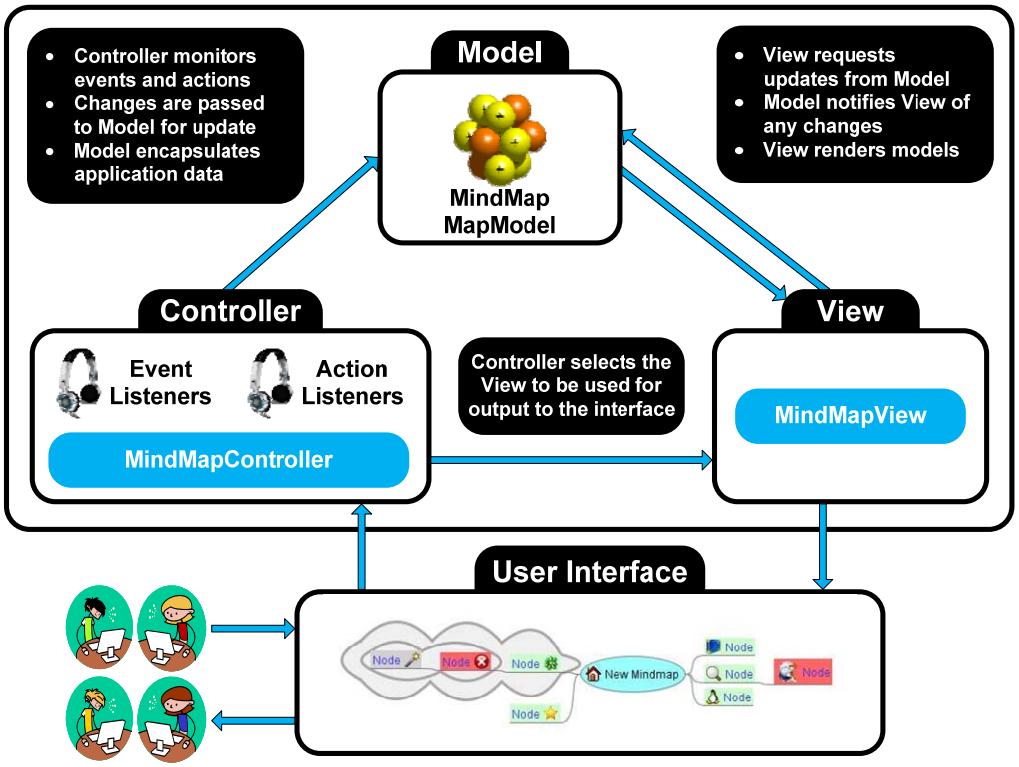


Figure 5-15 Freemind Architecture

By decoupling Model and View, MVC helps to reduce the complexity in architectural design, and increases the flexibility of components while enabling the reuse of models and views. As Controller tends to be application specific, there is limited scope for external reuse. Java Swing, which is used to implement Freemind, does not necessarily use a single instance of Controller. As the Swing event model is based on Java interfaces, it is common for applications to create an action class for each event. The Controller is then contained within the Event dispatching thread, which captures and propagates events to the View and Model, as illustrated in Figure 5-15 above.

The modular design of Freemind means that potentially the system can be used to edit data from various data sources. To make specific data available for use in Freemind, a new "mode" has to be written for that data source. U-CREATe implements only one mode of Freemind, Mind mapMode. However, other modes can be developed and integrated in the system. Freemind currently has a trial version of other modes, FileMode and BrowseMode available. FileMode is used to represent an operating system file tree as a Mind map. Data, behavior, node style, edge style, color, etc. are all controlled and defined by the mode.

5.3.3 User Interface

Freemind provides a feature-rich interface for the creation, editing and browsing of Mind maps. A screenshot of the interface is presented in Figure 5-16 below. A Mind map consists of information stored in various graphical text boxes called nodes. Nodes are connected using lines called edges. There are multiple ways to navigate around a map in Freemind, including drag and drop functionality for moving nodes and re-positioning the map. Horizontal and vertical scrolling are also available.

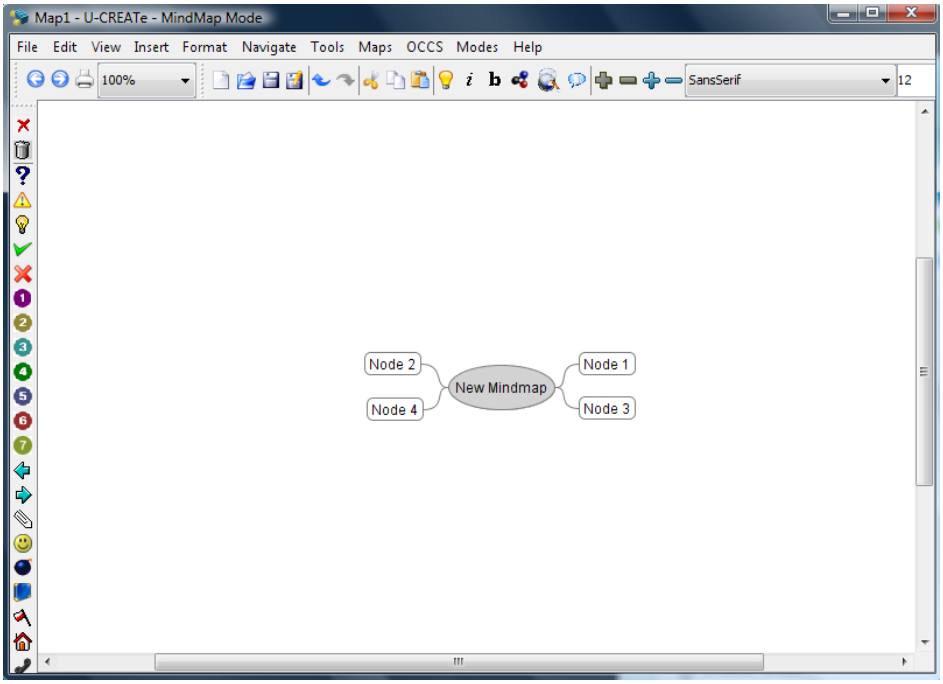


Figure 5-16 Freemind User Interface

Clicking on a node causes all child nodes to fold, so they are hidden from view below the clicked-on node. You can drag and drop single or multiple selected nodes in a map from one location to another. In addition, you can drag and drop text or lists of files from outside applications into Freemind.

There are many graphical features to alter the appearance of Mind maps. Nodes can have different colours for both text and background. Text can also have different styles, such as bold and italic, sizes and fonts. Different graphical styles can also be applied to a node, such as bubbled and forked. Nodes can contain anything from a single word to several paragraphs of text. Such features are demonstrated in Figure 5-17 below.

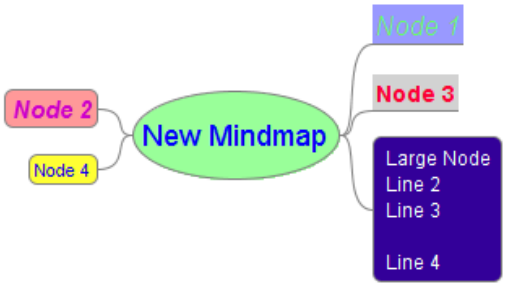


Figure 5-17 Freemind Node and Text Styles

Icons can be added to nodes. This can be used to differentiate between node types or to highlight nodes that contain certain features. Clouds can be used to visually group nodes. The appearance of edges can also be manipulated and various styles applied. Edge style options include Linear, Bezier, Sharp Linear and Sharp Bezier. Edge colour and size can also be altered. Other physical styles for both nodes and edges can be user-defined by adding to an XML file that describes all the styles used in Freemind. Graphical links can be added between nodes. This can be used to show an association between nodes in different hierarchical branches or between sibling nodes within a single branch. Such features are demonstrated in Figure 5-18 below.

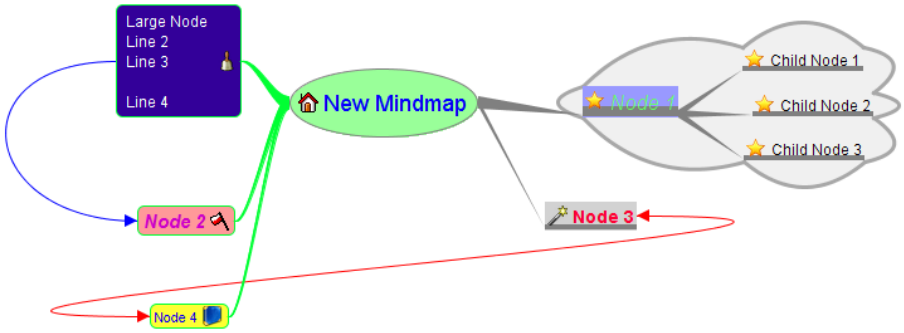


Figure 5-18 Freemind Node and Text Styles

Mind maps can be exported in various formats for incorporation into other applications. There are export options for HTML, XHTML, PNG, JPEG, SVG, XSLT, and OpenOffice. It is also possible to import files from other Mind map systems such as MindManager. It is possible to cut and paste Mind maps or branches of a map into Microsoft Word, Wordpad or Microsoft Outlook. It should be possible to paste a Mind map into any application that understands rich text format.

It is possible to use HTML tags to manually edit the appearance of a node. Tags can be used to edit the style of text or to insert formatting controls such as tables or linked lists. Images can also be attached to nodes. Supported image formats are PNG, JPEG and GIF. However upon inserting an image into a node, any text in that node will be lost. It is also possible to include an image in a node through the use of HTML <img> tags. The inclusion of an image in a Mind map is demonstrated in Figure 5-19 below.

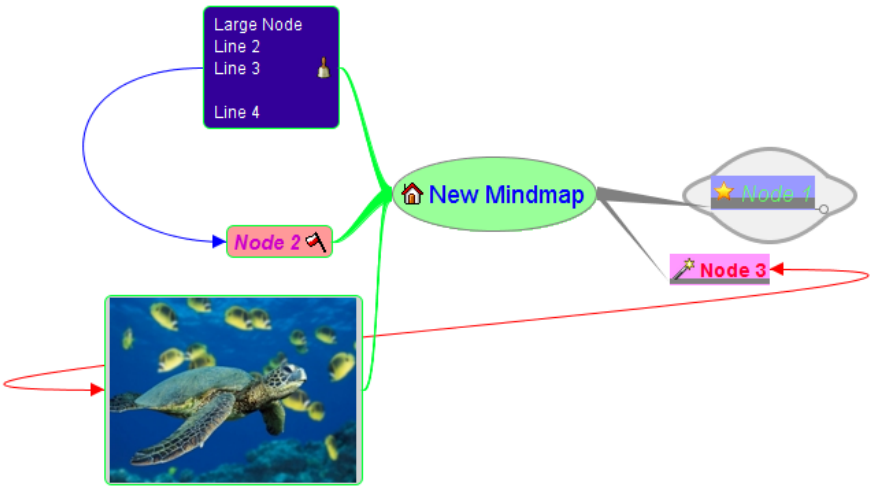


Figure 5-19 Freemind Node and Text Styles

It should be noted that there is currently no support for HTML formatted nodes or pictures when exporting to text or RTF. However, this method of stylising a Mind map can still be convenient for use on the Web. A detailed User Manual which describes the operation of U-CREATe is available [Appendix D], and a list of frequently asked questions can be found at [U-CREATe FAQ].

5.3.3.1 OCCS Integration

U-CREATe builds upon the Freemind interface and extends it in two main ways. The manner in which links can be added to a node in a Mind map has been completely re-written in U-CREATe. This was an extremely complex implementation task and required a significant amount of code changes. This was implemented to satisfy U-CREATe requirement **vi**. The second main change is that OCCS accessibility has been built into the U-CREATe interface using a variety of navigation options, this functionality was implemented to satisfy U-CREATe requirements **iii**, **iv** and **v**.

U-CREATe offers a variety of options for linking a node in a Mind map to external content. It is possible to create links to web pages, e-mail addresses, executables, local folders or any files on a local computer or network. As a learner undertakes the process of describing their cognitive model of a topic through the creation of a Mind map, the OCCS can provide much needed support. The learner can search across a subject-specific content cache for content which describes individual concepts within their map or the subject area as a whole. The OCCS search interface can be launched directly from a U-CREATe Mind map. The OCCS can be launched via a menu-bar icon (Figure 5-20), a drop-down menu (Figure 5-21) or a right-click menu (Figure 5-22).

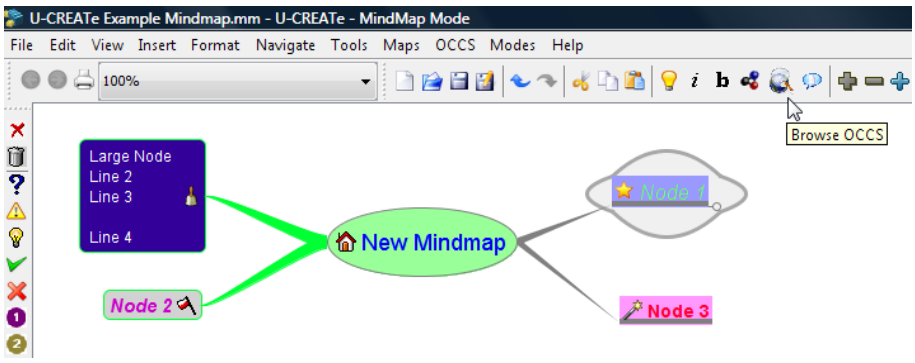


Figure 5-20 U-CREATe Menu-Bar OCCS Launch Icon

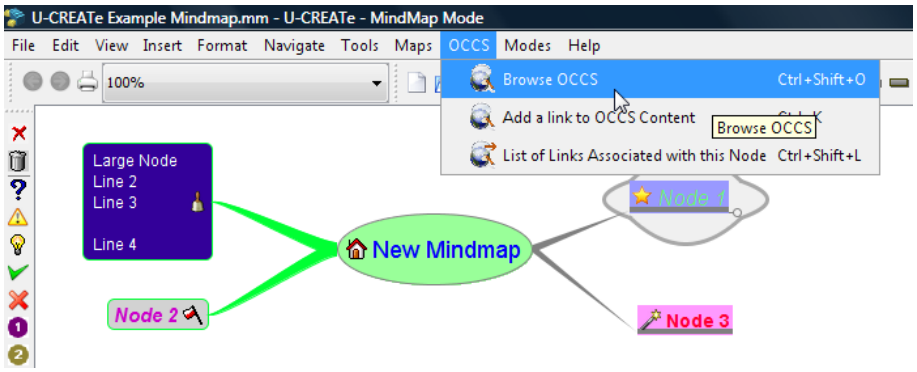


Figure 5-21 U-CREATe OCCS Menu Options

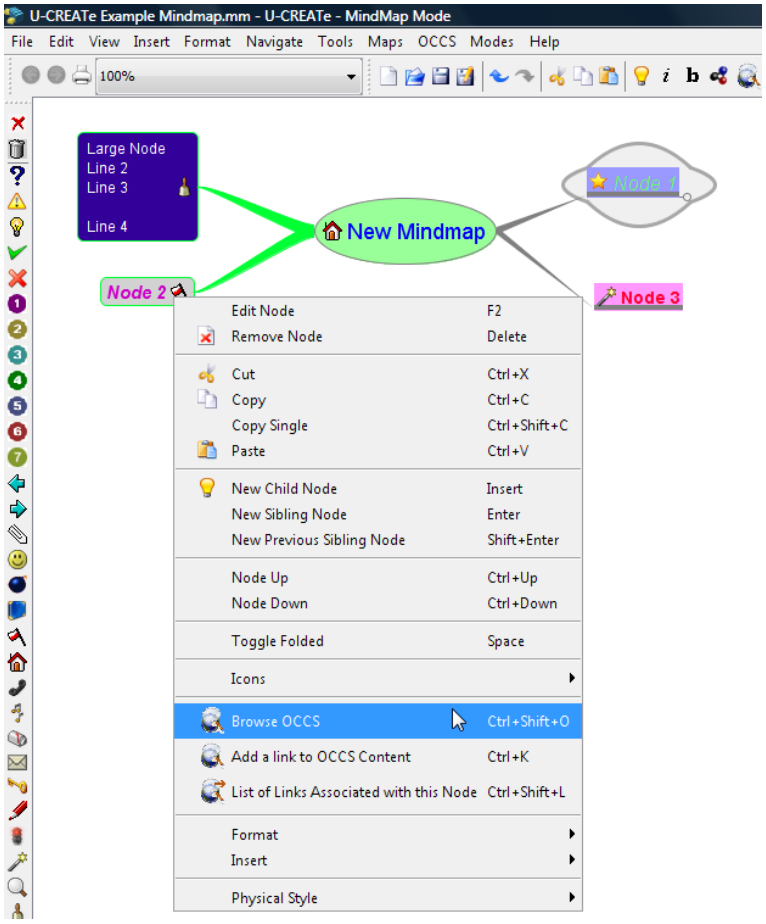


Figure 5-22 U-CREATe Right-Click Menu - OCCS Options

When any of these OCCS links is clicked, the web-based search interface is launched in a separate browser window. This allows the learner to continue viewing and modifying their mind map while they search for related content on the OCCS. This was necessary to satisfy requirement **v**.

As the learner identifies quality pieces of content related to concepts within their Mind map, links can be attached to the nodes in question. Multiple links can be added to a single node, as illustrated in Figure 5-23. This enables learners to associate numerous pieces of content in the OCCS with each concept in their Mind map. Various icons are used to indicate links which are contained within a node. The icon used is dependent upon the type of links attached to the node. It is possible to browse the links associated with a node and view the content of each OCCS link in a browser window. Links can be easily removed and added at any point.

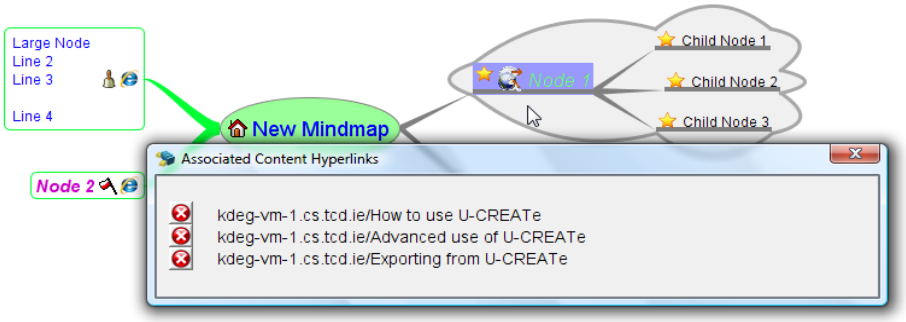


Figure 5-23 U-CREATe – Node Links List

This adds a knowledge layer to the Mind map and increases the educational value of the completed map. The OCCS in conjunction with U-CREATe can act as a scaffold to help the learner plot their map and can guide the process of knowledge acquisition as they expand their understanding of a topic.

The process of searching the OCCS for appropriate learning content for each concept, and filtering the results to identify the most applicable information constitutes an Enquiry Based Learning experience. Such experiences are defined by the active participation of learners in an investigative process within a subject domain. The ability to deliver a pedagogically meaningful educational experience that is supported through the provision of content from open corpus sources constitutes a valuable and innovative educational tool.

5.4 Summary

Chapter four of this thesis defined a series of technical requirements for the design of both a content discovery and retrieval service for TEL and a method of exploring and integrating such open corpus content within an educational experience. This chapter described the implementation of the Open Corpus Content Service (OCCS), which delivers methods of content discovery, harvesting, classification, indexing and delivery. It also described the implementation of the User-driven Content Retrieval, Exploration and Assembly Toolkit for eLearning (U-CREATe), which delivers an interface for the learner-driven creation of Mind maps and the ability to explore and reuse educational content provided by the OCCS.

A rapid prototyping approach was used to develop the methods and components of both the OCCS and U-CREATe. In parallel a series of mini-evaluations lead to the further refinement of the prototypes. This chapter provided a detailed insight into the architecture of the OCCS

which is based upon the Information Retrieval techniques of focused crawling and text classification described in chapter three. The components of the OCCS were described and illustrated in this chapter. Also described here is the implementation of U-CREATe, a TEL application whose development has been guided and informed by the educational theories described in chapter two. A description of the implementation of U-CREATe is presented. This chapter concluded with an examination of the functionality of U-CREATe and its integration with the OCCS.

6 Evaluation of the OCCS and U-CREATe

6.1 Introduction

The two main objectives of this research were derived from the overall research question which was posed in chapter one of this thesis. These objectives were *(i) to investigate, prototype and evaluate an approach for supporting the discovery, classification, harvesting and delivery of educational content from open corpus sources* and *(ii) to design, prototype and evaluate a TEL application which enables the incorporation and resource-level reuse of such content during the execution of a pedagogically meaningful educational experience*. Two applications, the OCCS and U-CREATe, were designed and developed which focus on these research objectives and aim to address the overall goals of this thesis. The aim of this chapter is to provide detail and analysis of experiments which evaluate the performance of the OCCS and U-CREATe.

The OCCS and U-CREATe applications combine to provide a foundation for the reuse of educational content in technology enhanced learning systems. The technical requirements for these two applications are specified in chapter four of this thesis and the technical implementation of the systems to meet those requirements is detailed in chapter five. In order to evaluate how well the research objectives stated above are achieved, two distinct, yet complimentary experiments were designed and conducted to assess the performance of both the OCCS and U-CREATe. The discovery and reuse of existing content is the key motivation for this research and, as such, both experiments were designed with this motivation in mind.

The first experiment is designed to evaluate the discovery, classification and harvesting of open corpus content. This experiment examines the performance of the OCCS in sourcing subject-specific content from the WWW. An experimental trial is presented which scrutinised the relevance, technical validity and quality of the content retrieved by the OCCS. This trial involved the execution of a focused web crawl and the subsequent examination, by subject-matter experts, of the subject-specific content harvested during the crawl.

The second experiment is designed to evaluate the educational performance of a TEL offering which implemented the resource-level reuse of open corpus content with the aim of highlighting the potential for open corpus content reuse in pedagogically beneficial

educational scenarios. This experiment examines the utilisation of a cache of open corpus content, generated by the OCCS, in a TEL offering delivered in U-CREATe. This involved a trial of U-CREATe with real learners in a defined educational scenario.

The experiments presented in this chapter were conducted through a series of trials with both subject-matter experts and undergraduate students. Both the OCCS and U-CREATe were developed using an iterative, rapid-prototyping approach. Presented in this chapter are the results of the most recent evaluations conducted.

The goal of this chapter is to detail these experiments and provide an examination and analysis of the results emerging from them. Section 6.2 details the experiment to examine the performance of the OCCS in the generation of caches of open corpus content. Section 6.3 describes the experiment to examine the performance of U-CREATe in the utilisation of such open corpus content caches. Both sections provide detail on the experiment objective, the evaluation metrics employed, the experiment methodology used and an analysis of the results which emerged.

6.2 Experiment One: The Discovery, Classification and Harvesting of Open Corpus Content from the WWW

6.2.1 Experiment Objective

A series of technical requirements for a novel information retrieval tool-chain were specified in section 4.3.2 and the OCCS was implemented to meet these requirements. The OCCS tool-chain is designed to be used by course developers and educators as a means of discovering and accessing content suitable for supplementing their TEL offerings. While the technical requirements can be deemed to be satisfied by an examination of the implementation of the OCCS in section 5.2, it was necessary to conduct an explicit examination of the performance of the OCCS with regard to the generation of subject-specific corpora to determine if the objectives of this thesis have been fully satisfied.

In order to assess the performance of the OCCS with regard to the generation of such corpora, the quality of the content identified and harvested needed to be scrutinised, both in terms of its relevance to the scope of the web crawl, and its suitability for inclusion in educational offerings. The individuals deemed most capable of performing such an

assessment were educators with expertise in the subject area in question and a familiarity with TEL.

This experiment aims to conduct such an assessment of the performance of the OCCS in generating caches of subject-specific content. The scope for a focused crawl is defined in conjunction with a domain expert and the crawl is conducted on the open WWW. The cache generated by this crawl is then examined and assessed by selected domain experts. The following sections detail the performance metrics used to evaluate the results of this experiment and the methodology used to conduct the trial. This is followed by a detailed examination of the experiment results and an analysis of the system performance.

6.2.2 Evaluation Metrics Employed

Many IR-based research projects target the Text REtrieval Conference (TREC) [TREC] for performing system evaluation. TREC was founded by the National Institute of Standards and Technology **(**NIST) [NIST] in 1992 to provide the research community with access to a very large test collection which had been developed for the DARPA TIPSTER project [Harman 92]. TREC is used to produce quantitative results to compare and contrast algorithmic IR approaches over several identical tasks on pre-defined corpora. However, it is not the goal of this experiment to evaluate the performance of the NutchWAX retrieval algorithm, this experiment aims to evaluate the performance of the OCCS in generating subject-specific caches of content from the WWW. As TREC requires the use of pre-defined corpora, it was deemed unsuitable for the purposes of this evaluation. The IR metrics used in this experiment are designed to evaluate the quality of the content in the subject specific cache, rather than the performance of the IR algorithm employed by the OCCS web-based interface.

Each of the trial participants were asked to rate the content encountered for relevance, technical validity and quality.

- Relevance is examined with relation to the scope of the focused web crawl. As learners will perform queries to satisfy a particular information need, the content returned must satisfy, or be relevant to, that information need. These relevance assessments are then used to calculate defined IR performance measures, which are described in more detail in the following paragraphs.

- The technical validity of the content is examined to ensure that the content retrieved is factual in nature and technically consistent. This is important to ascertain, as if technically inconsistent or factually incorrect material was included in an educational offering, it could potentially be detrimental to the learner.
- Quality is examined to ensure that the harvested material which is being returned in the results lists is of sufficient quality to be suitable for inclusion in a TEL offering within a defined context. Educators would not wish to include content in their educational offerings which was of poor quality.

Six Information Retrieval performance measures are used to assess the performance of the OCCS based upon the domain experts' relevance assessments. There are four measures calculated for each query performed: Precision, Average Precision, Reciprocal Rank and Total Reciprocal Document Rank; and two measures of overall system performance: Mean Average Precision, and Mean Reciprocal Rank. The majority of common retrieval evaluation methods require each document to be rated as either relevant, or not relevant. This can be an invalid assumption, as depending on the target audience, the information need and the query, there may be different levels of relevance and therefore, this evaluation has asked the domain experts to rate relevance on a ten point Likert scale. However, in order to enable the calculation of these IR performance metrics, a result is deemed relevant if it receives a relevance rating of $5.1/10$ or above.

The two most common IR performance measures are Precision and Recall. Precision measures the percentage of results returned for a query which are relevant. Recall measures the percentage of all possible relevant results which are returned for a query. It is not possible to measure recall on the open WWW as the complete set of relevant results for a given query is not known [Chakrabarti et al. 99]. However it is possible, and valuable, to measure Precision when searching across web content. Another common measure of performance is van Rijsbergen's F_1-measure [van Rijsbergen 79], which calculates the weighted harmonic mean of precision and recall. However as you cannot calculate recall on web content, the F_1-measure also cannot be calculated.

Whilst the precision measure for a set of results is valuable, it does not take into account the importance of ranking in modern, web-based, information retrieval. The average precision is

a measure used to estimate the performance of an IR system by placing emphasis on relevant results appearing high up the ranked list. Average precision is calculated by finding the average of the individual precision calculations for each relevant result in the list, if the list were truncated directly after that result.

$$AP = \frac{\sum_{r}^{N}(P(r) * rel(r))}{rd}$$

Where r is the document at the current rank, N is the number of documents retrieved, $rel(\ )$ is a binary function on the relevance of r, $P(\ )$ is the precision given a cut-off point of r and rd is the number of relevant documents returned. The Mean Average Precision (MAP) for the IR system is then calculated by summing the average precision value for each query conducted and dividing by the total number of queries. This gives an indication of the overall performance of the system with relation to precision and ranking.

The reciprocal rank of a results list is the rank at which the first correct answer occurs. When calculating the reciprocal rank, a value of $1/1$ is assigned if the 1st result is relevant, $1/2$ if the 2nd result is the first relevant result and $1/3$ if the 3rd result is the first relevant result, and so on. The Mean Reciprocal Rank (MRR) is the average rank for the IR system at which the first correct answer occurs. This score has been used in TREC to evaluate the retrieval performance of IR approaches. The average reciprocal rank assigned by the evaluation participants for each query are averaged to find the MRR for that query [Voorhees 99] [Roussinov et al. 08].

Total Reciprocal Document Rank (TRDR) [Radev et al. 05] is the sum of the reciprocal values of the ranks of each of the relevant results from among the top n results provided by the system. In this case n is ten, as the participants are examining the top ten results for each query. For example, if a query is performed and the third, fourth, eighth and ninth results are deemed to be relevant, the TRDR for that result is calculated as follows:

$$1/3 + 1/4 + 1/8 + 1/9 = 0.819$$

The maximum achievable TRDR value varies dependent upon how many results are being examined. In the case of this evaluation, the maximum achievable value is 2.93. TRDR allows for more subtle measures of performance than MRR, as TRDR takes into account all the results returned by the system which have been rated for relevance, rather than just the

first occurrence of a relevant result. For instance, consider two result sets, in one only the third result is relevant and in the other the third, fourth, fifth, sixth, seventh and eight results are all relevant. Both of these would be assigned the same MRR, but different TRDR values.

6.2.3 Experiment Methodology

A focused web crawl was conducted to generate a content cache using the OCCS. For the purpose of this evaluation, the scope of the crawl was the SQL programming language. This decision was taken for a number of reasons. Firstly, there were a number of domain experts in this area available within the Department of Computer Science in Trinity College Dublin, who could evaluate the content harvested by the OCCS. Secondly, there was an appropriate undergraduate class within the same department who were involved in the study of SQL and could be targeted for participation in the U-CREATe evaluation which is detailed in section 6.4. Additionally, the research group of which I am a member has an existing adaptive hypermedia course in SQL, which could be used in future comparative experiments.

Once the crawl had been completed and a corpus of content created by the OCCS, experiment participants were required to evaluate the relevance, technical validity and quality of the content. Five participants were identified in the School of Computer Science and Statistics at Trinity College Dublin who met the necessary requirements. They were each experts in the domain of SQL and were each familiar with TEL.

To conduct this examination of the content cache it was necessary to create a method of assessment whereby these selected domain experts could semantically analyse and critique the content. When dealing with such a large volume of content it was infeasible to manually examine the entire cache. It was decided to examine the content using the manner by which most learners would interact with the cache. A web-based interface was implemented, where selected domain experts could perform specific search queries on the index of the content cache. These searches were conducted in the context of a pre-defined intended audience. This intended audience was defined as a class of "Undergraduate Computer Science Students, with limited knowledge of the SQL programming language but with some technical computing knowledge". The participants were provided with the following introductory information.

"*The content upon which you can search in the OCCS was sourced from the WWW and is related to the SQL Programming Language.*

When you perform a search, the content should be rated for its applicability to the query, within the scope of the SQL Programming Language. There will be some content in the cache which is out of scope e.g. install instructions for particular database types, if returned, this content should be marked as irrelevant.

For each query performed you need to click on each of the top ten results and select a value from the three Likert scales available on-screen."

Seven search queries were selected to be performed on the content index which were intended to be typical of such a student's information needs. The queries were designed to evaluate both the concept coverage of the content cache generated by the OCCS and the performance of the query-document similarity calculations of the search interface. These queries have been divided into two sets. The queries in set A, queries one to four, are searches for technical SQL concepts related to the syntax of the programming language which are covered by elements contained in the keyword file and are therefore within the scope of the web crawl. The queries in set B, queries five, six and seven, are searches for concepts which are directly related, yet peripheral, to the SQL programming language. These concepts are not covered by elements in the keyword file and therefore are not within the scope of the web crawl. These queries examine content coverage at the crawl boundary. Query seven also examines query structure and stop-word removal.

The queries were:

- Query Set A
    - SQL Select Statement
        - A search to find content which would provide an introduction to the concept of the SQL select statement.
    - SQL Insert Statement
        - A search to find content which would provide an introduction to the concept of the SQL insert statement.
    - SQL "Create View"

        - A search to find content which would provide an introduction to the concept of Views and the SQL "CREATE View" statement in particular.
    - Foreign Key
        - A search to find content which would provide an introduction to the concept of Foreign Keys.
- Query Set B
    - Schema
        - A search to find content which would provide an introduction to the concept of database schemas.
    - Relational Databases
        - A search to find content which would provide an introduction to the concept of Relational Databases.
    - What is a Stored Procedure
        - A search to find content which would answer a specific question. In this case information on the concept of stored procedures.

Each participant performed all seven queries and examined the top ten results returned by the OCCS for each of these queries. Each participant was then required to rate each of these search results for relevance, technical validity and quality by answering questions provided in the web interface. These questions and the associated notes provided to the participants were:

- How relevant is this result to the query performed?
    - Is the content contained within this result relevant to the information need for which the query was conducted.
- Is this result factual in nature?
    - Are there any technical inconsistencies / irregularities?
- What is the overall quality of this result?
    - The content is to be used in an undergrad class with students who have begun to learn about SQL but are relative beginners. Is the content of sufficient quality for this context?

An input area was also provided for each participant to enter any additional comments that they may have regarding each query, or the evaluation in general, and participants were asked

if they had encountered any non-english language content. The questions are answered using a Likert scale. This is a type of psychometric response scale used in questionnaires, and is the most widely used scale in survey research. When responding to a Likert questionnaire item, respondents specify their level of agreement to a statement [Likert 32]. Having discussed this option with Mary Sharp, a data analysis expert from the Department of Computer Science in Trinity College Dublin, it was decided to use a ten point Likert scale, as this provides no midpoint on the scale. Individuals will tend to err on the side of caution and choose a neutral response if one is provided.

6.2.4 Content Sourcing

This experiment began with the execution of a focused web crawl by the OCCS. As described in section 5.2.10, the process of training the rainbow classifier defines the scope of the content which will be discovered and harvested by a focused web crawl conducted using the OCCS. There are three user-defined input files which guide this training process, the keywords file which describes the subject area, the positive ODP category file which defines areas of the ODP taxonomy which fall within the scope of the crawl and the configuration file. The versions of these files used in this evaluation can be found in Appendix E. Figure 6-1 opposite, illustrates the process flow involved in conducting this experiment, from classifier training through to crawl execution and cache generation.

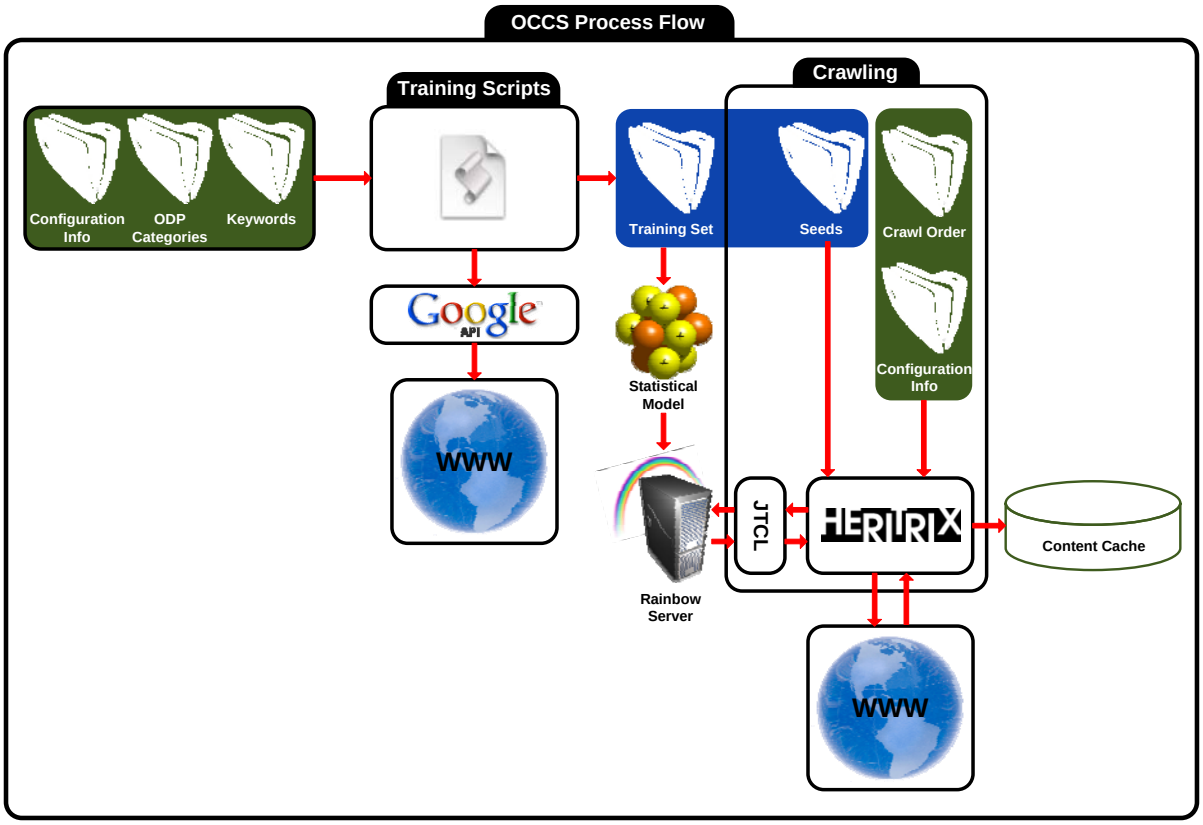


Figure 6-1 OCCS Content Cache Generation – Process Flow

The keywords file has a direct influence on the classification model generated by rainbow and as such, has a significant impact on the content that is deemed relevant to scope and accepted into the cache. For every keyword provided, a query is sent to the Google API and a defined number of the results returned are added to rainbow's training set as positive domain examples. This number is specified as a configuration option and was varied through iterative experimental crawls. For the purposes of this evaluation the three top-ranked results from each query were added to rainbow's positive training set. The top result for each query is also added to the list of seed URIs for the crawl.

Over the course of this research various approaches have been taken to the generation of the keywords list. All of these approaches involved academic domain experts defining the scope of the subject area. In an initial evaluation each educator was provided with the index of terms from the well know database reference book, C.J. Date's "An Introduction to Database Systems" [Date 99] and asked to select all the terms which they felt fell within the scope of SQL. The selections of each participant were then merged into one keyword list. This list was trimmed down in conjunction with one of the subject-matter experts to remove duplication

and to ensure accuracy to scope. For the evaluation described in this chapter, an ontology describing the area of SQL was defined in consultation with domain experts. This ontology logically divided the subject area into programmatic commands and conceptual elements. This ontology can be found in Appendix E. Each of the base hierarchical elements in this ontology was combined with contextual terms for general purpose web search, such as "SQL" and "Command", and entered as keywords for use in the rainbow training process.

The positive categories file contains a list of all the ODP categories which fall within the scope of the crawl. The ODP category hierarchy can be manually browsed via the DMOZ website. One category was deemed to be specifically applicable for this evaluation, it was "Top: Computers: Programming: Languages: SQL". All of the content contained within this category was extracted and added to rainbow's positive training set. This file is also used to generate rainbow's negative training set. The set difference between the universal set of ODP categories and the set of positive ODP categories is found and a defined number of random categories from this set are added to a negative category list. For the purposes of this evaluation, this configuration option was set to 200. Once this list of negative categories has been generated, the content contained within the categories is extracted and added to rainbow's negative training set.

The rainbow relevance boundary also plays a key role in the generation of a content cache by the OCCS and is set in the configuration file. This boundary can be assigned any value on a scale from 0 to 1. For instance, a boundary of 0.7 would mean that content must be deemed to be greater than, or equal to, 70% accurate to the rainbow classification model to be included in the content cache. The setting of this boundary forms a balance between domain coverage and cache dilution. If the boundary is too high, some potentially relevant content could be excluded from the cache. If it is too low, the cache may become diluted with irrelevant content. Iterative crawls were conducted using various boundaries to ascertain the most beneficial level. For the purposes of this evaluation the relevance boundary was set at 0.9 or 90% relevance.

The seed file generated by the rainbow training process was manually reviewed to ensure accuracy to scope. The final version of the file contained 186 seeds which provided the points from which the crawler began its traversal of the web. This seed file can be found in

Appendix E. A second file is produced by the rainbow training process which lists the N most frequent terms which occur in the classification model. The number of terms generated is a configuration option, and for the purposes of this evaluation was set to 200. This file essentially acts as a stop-word list. Terms can be marked as stop words if they are deemed to be too general, too common or occur too frequently to be of use in content discrimination. This list was examined with the aid of a domain expert and words that were identified as being of low value were tagged for removal. This process was repeated until the top words file contained only terms which were of use in the classification process. Once this fine tuning process was complete, the rainbow model was re-generated and rainbow launched as a service, listening on a defined port.

The focused crawl was then initiated using a BroadScope and without defining termination parameters as the crawl was going to be monitored manually. During each crawl Heritrix monitors and logs details regarding the number of URIs which have been discovered, queued for download and downloaded. These statistics can be viewed in figures 6-2, 6-3 and 6-4 below. The number of URIs discovered and queued for download starts off quite high as the crawler processes the seed URIs which are all of high relevance and good sources of content. There are also six distinct peaks in both the discovery and queuing graphs. These could be attributed to the identification of rich content hubs, sources of URIs which were being exploited during those periods. The overall performance of discovery and queuing deteriorate slightly over the course of the crawl. This is somewhat expected, as discovered paths to relevant content become fully explored and new content becomes less readily discoverable. However this can also be partly attributed to inefficient memory management in the crawler, which also somewhat impedes the performance of URI download in the latter stages of a crawl. This is discussed further in section 6.2.4.1.

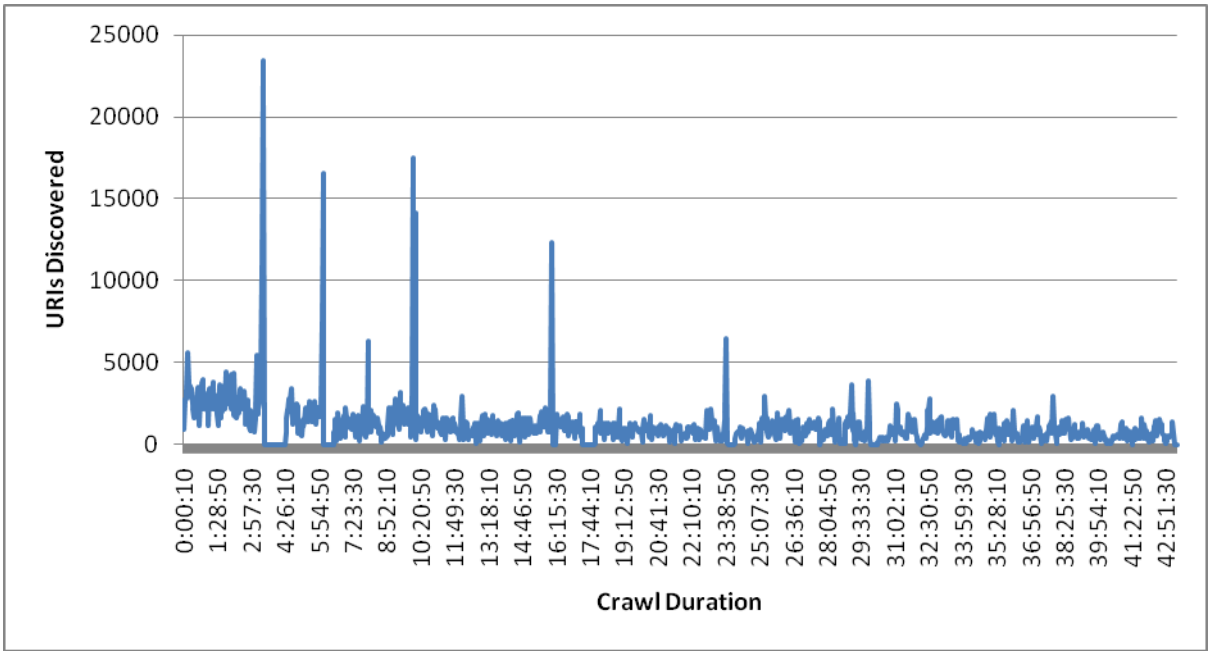


Figure 6-2 URI Discovery over Crawl Duration

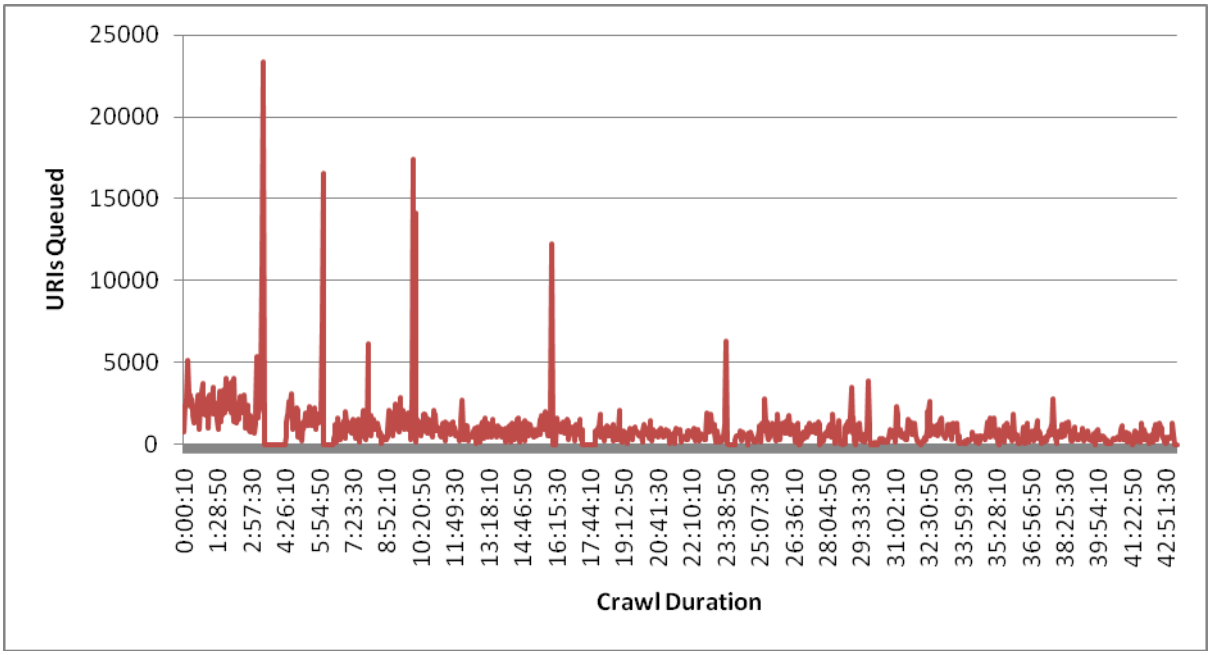


Figure 6-3 URI Queuing over Crawl Duration

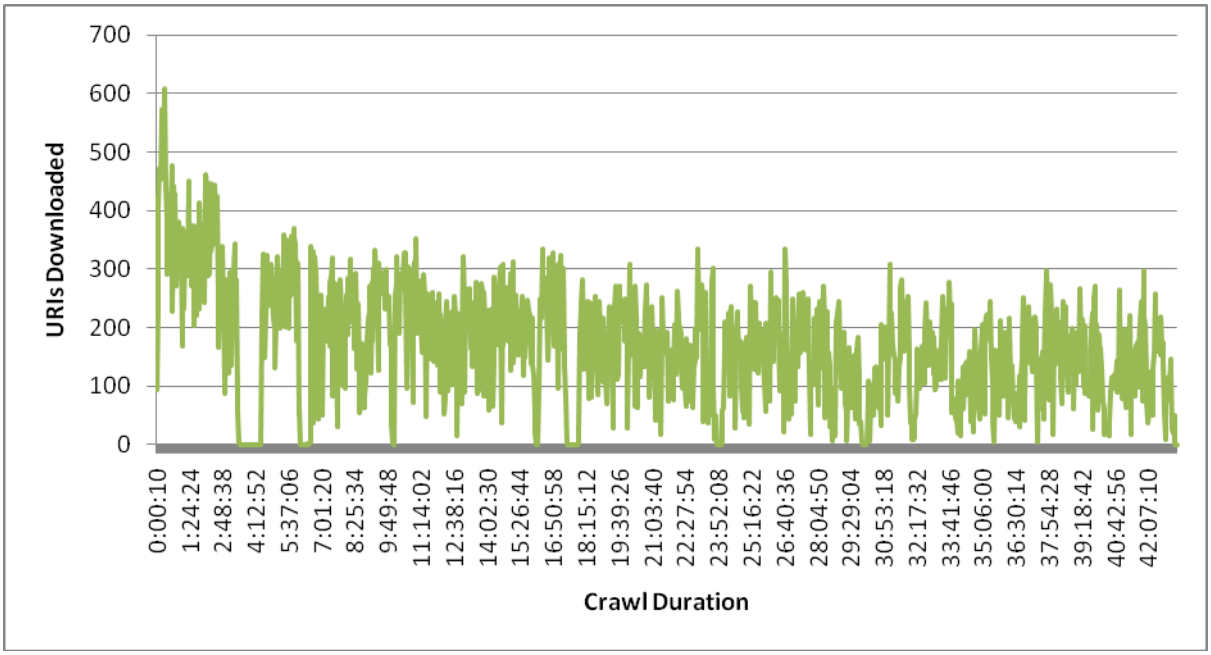


Figure 6-4 URI Download over Crawl Duration

The crawl ran for a total of 43 hours, 33 minutes and 53 seconds, at which point it was manually terminated using the web interface. The URI statistics for the crawl are illustrated in Figure 6-5 below. At termination Heritrix had discovered 1,361,489 URIs with 1,145,444 scheduled for download. Of these 201,626 were downloaded and passed to TextCat for language classification. TextCat labelled 191,517 URIs as "English" and passed them to the rainbow classifier. Rainbow adjudged 149,933 results to be above the 90% relevance threshold and these were duly included in the content cache. 41,584 URIs were classified by rainbow as being below the 90% threshold and as such they were disregarded.

URIs	Discovered	Queued	Processed by TextCat	Processed by Rainbow	Entered into Content Cache
Crawler Finished	1,361,489	1,145,444	201,626	191,517	149,933

Figure 6-5 Overall Crawl Statistics

6.2.4.1 Key Observations and Considerations

The configuration of individual crawls can take some familiarisation on the part of the system administrator. However, the configurability of Heritrix means the system is very flexible and crawls can be manipulated to suit exact project requirements. The inputs required from the educator for each crawl are very simple and non-technical in nature. They need only provide a file containing keywords which describe the subject area and ODP categories which fall within the scope of the crawl. There is also a need for them to review the seed file and stop-word list for relevance. At the moment this is implemented in a crude, manual manner which would not be sufficient for widespread use in mainstream education. Before this could take place, the provision of these inputs and the review of the training outputs will need to be re-implemented so it is more intuitive to the non-technical user, see section 7.4 Future Work.

Over the course of a crawl the efficiency of the OCCS software with regard to memory use tends to deteriorate. There is a tendency for the crawler to exhaust java heap space on the server at periods of heavy load, which is demonstrated in figures 6-2, 6-3 and 6-4 by the level of processing in discovery, queuing and download simultaneously moving to zero. While the crawler was running on a single virtual machine without major resources, it does seem to be reasonably heavy on memory when executing long, broadscope crawls. This did not cause a significant impact as, for the purposes of this research, we have generally run smaller scale, increasingly subject focused crawls. However, while this has not currently hindered OCCS crawls in a major fashion, the efficiency of the crawler will need to be examined and improved in future developments.

The OCCS has run on a virtual server for all the crawls conducted to-date. This virtual machine has 261Mb of allocated RAM, uses a 2.00GHz Intel Pentium 4 processor and has 30GB of allocated hard drive space. For future, larger scale crawls more memory and disk space can be allocated to the virtual server. The opportunity also exists to run the crawler on the National Computing Grid [Grid-Ireland], which is managed by the Computer Architecture Group in Trinity College and has enormous resources in terms of memory and bandwidth which could be exploited.

6.2.5 Content Assessment and Evaluation

The following section presents the results of the experiment conducted to evaluate the content cache generated by the OCCS during the focused crawl described in the previous section. A series of graphs are presented for each query conducted. These graphs contain the values entered for each search result by the domain experts who participated in the experiment. There are also graphs which detail which results were deemed relevant for each query. Accompanying these graphs are six Information Retrieval performance measures. There are four measures for each query performed: Precision, Average Precision, Reciprocal Rank and Total Reciprocal Document Rank (TRDR); and two measures of overall system performance: Mean Average Precision (MAP), and Mean Reciprocal Rank (MRR).

6.2.5.1 Content Relevance

The first, and arguably the most important, criterion that each of the evaluation participants were asked to rate the results list for was relevance. These ratings determined if the content returned was appropriate to the information need of the target audience. Relevance is the basis of the majority of traditional IR measures and as such was the key assessment metric for this evaluation. The following set of figures (Figures 6-6, 6-8, 6-10, 6-12, 6-14, 6-16, 6-18) display the relevance values assigned by the participants to each of the top ten results for each query performed in both query set A and query set B. There is also an associated figure for each query which displays the mean of these relevance ratings. The ranks at which index pages appeared in the results lists are indicated with a red circle on each of the figures. The significance of index pages and their impact upon relevance assessment is discussed in more detail in section 6.2.5.2. Accompanying these figures are the values for the four IR performance measures described above (Figures 6-7, 6-9, 6-11, 6-13, 6-15, 6-17, 6-19). The MAP and MRR values for the overall performance of the OCCS across the queries are presented at the end of this section.

Query Set - A

Query 1: SQL Select Statement

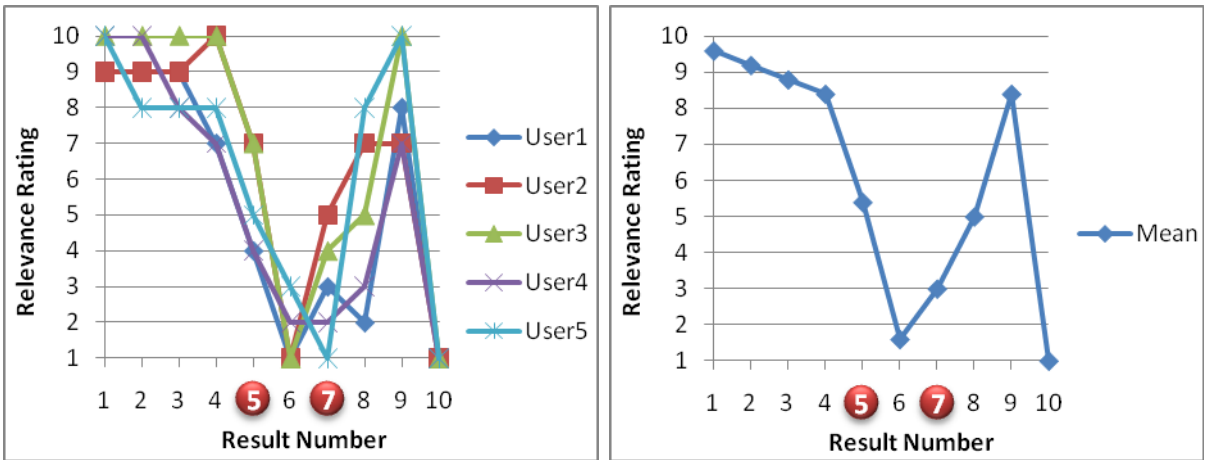
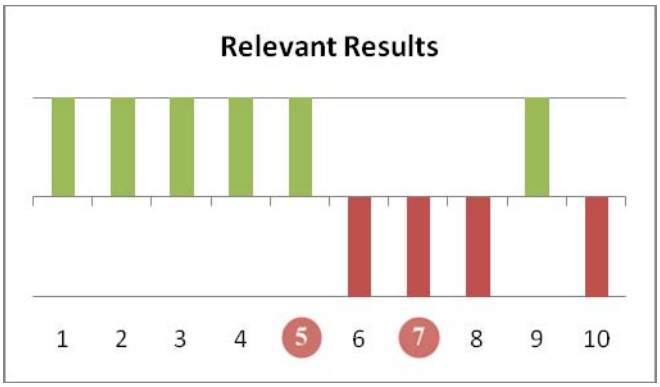


Figure 6-6 Query One Relevance Rating



Precision	0.60
Average Precision	0.94
Reciprocal Rank	1.00
Total Reciprocal Document Rank	2.39
Max Possible TRDR	2.93

Figure 6-7 Query One Performance Measures

Query 2: SQL Insert Statement

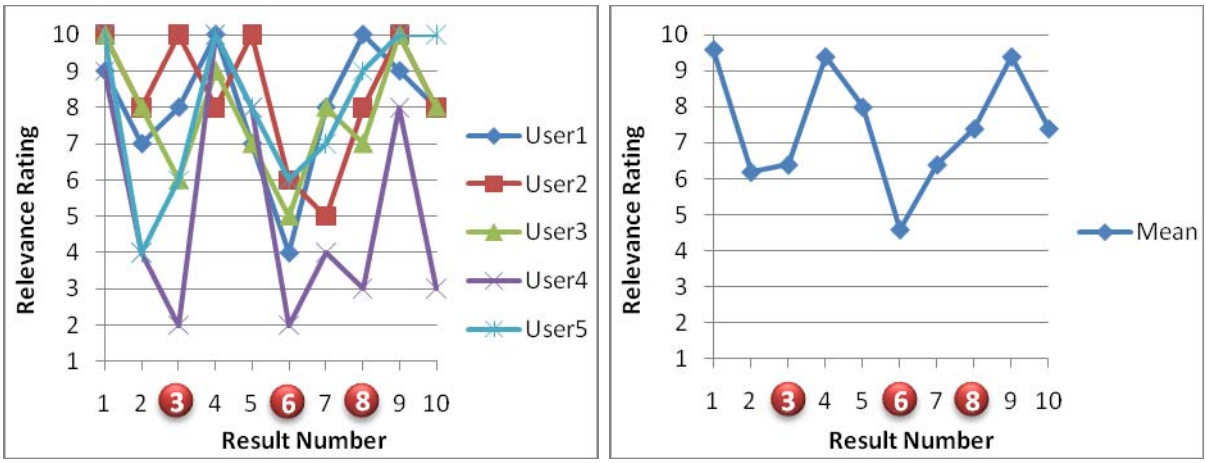
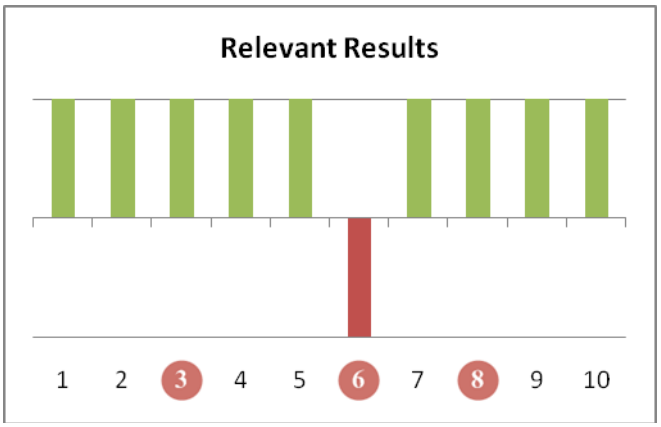


Figure 6-8 Query Two Relevance Rating



Precision	0.90
Average Precision	0.95
Reciprocal Rank	1.00
Total Reciprocal Document Rank	2.76
Max Possible TRDR	2.93

Figure 6-9 Query Two Performance Measures

Query 3: SQL “Create View”

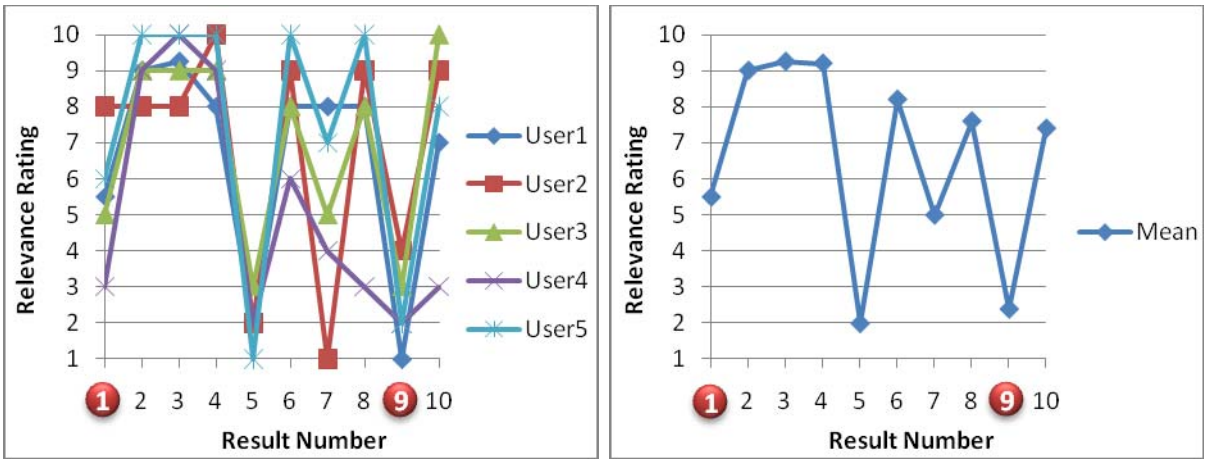
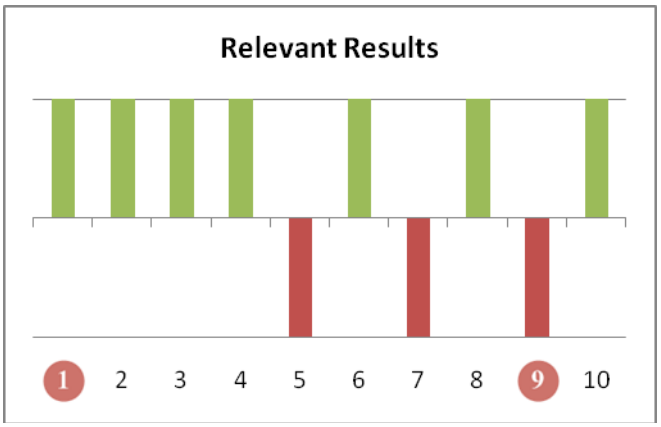


Figure 6-10 Query Three Relevance Rating



Precision	0.70
Average Precision	0.90
Reciprocal Rank	1.00
Total Reciprocal Document Rank	2.48
Max Possible TRDR	2.93

Figure 6-11 Query Three Performance Measures

Query 4: Foreign Key

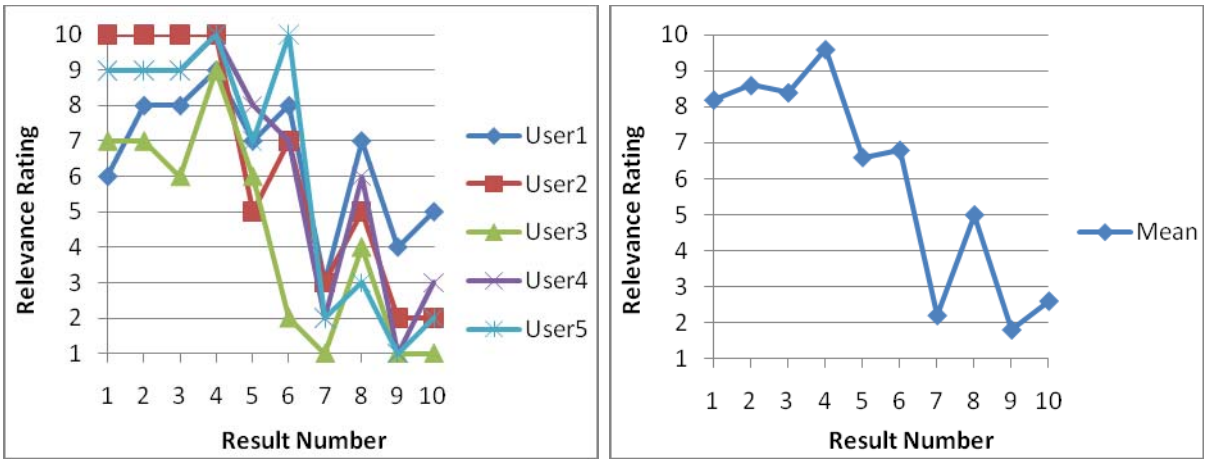
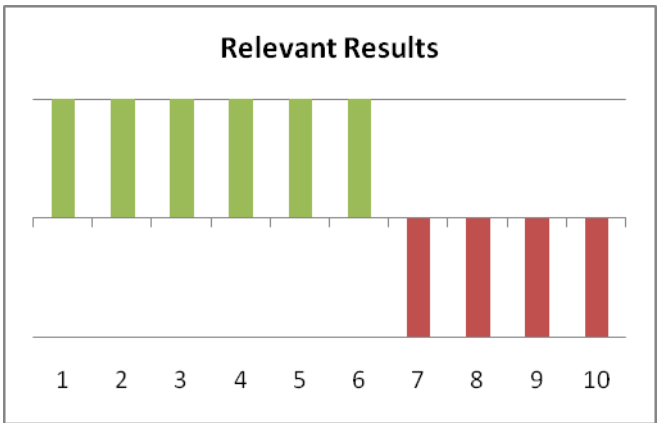


Figure 6-12 Query Four Relevance Rating



Precision	0.60
Average Precision	1.00
Reciprocal Rank	1.00
Total Reciprocal Document Rank	2.45
Max Possible TRDR	2.93

Figure 6-13 Query Four Performance Measures

Query Set - B

Query 5: Schema

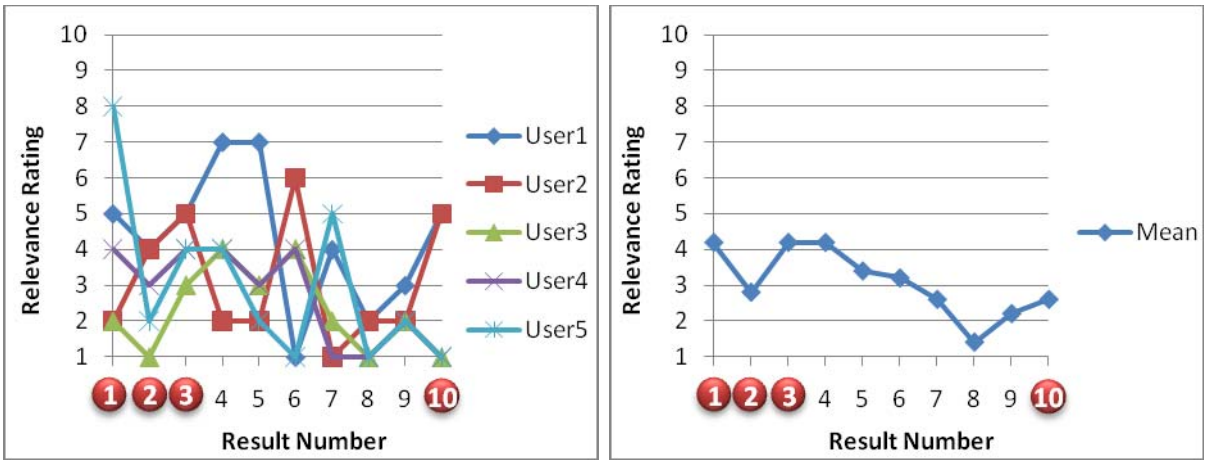
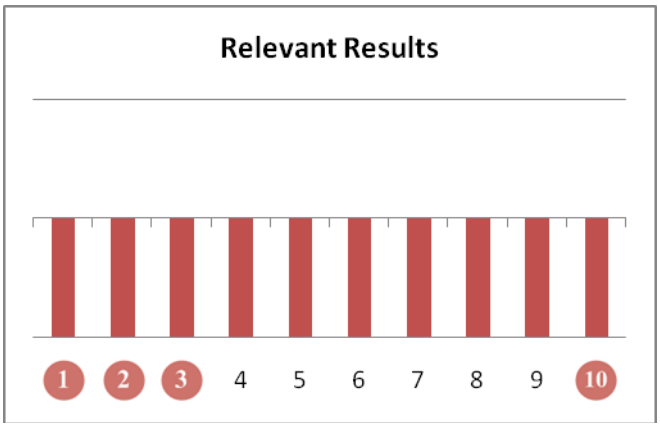


Figure 6-14 Query Five Relevance Rating



Precision	0.00
Average Precision	0.00
Reciprocal Rank	0.00
Total Reciprocal Document Rank	0.00
Max Possible TRDR	2.93

Figure 6-15 Query Five Performance Measures

Query 6: Relational Databases

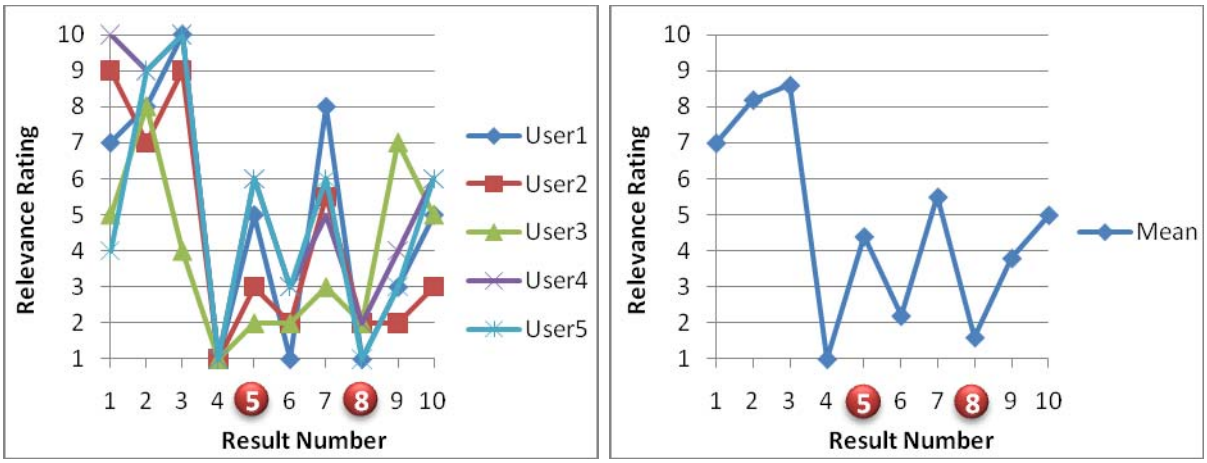


Figure 6-16 Query Six Relevance Rating

Precision	0.40
Average Precision	0.89
Reciprocal Rank	1.00
Total Reciprocal Document Rank	1.98
Max Possible TRDR	2.93

Figure 6-17 Query Six Performance Measures

Query 7: What is a Stored Procedure

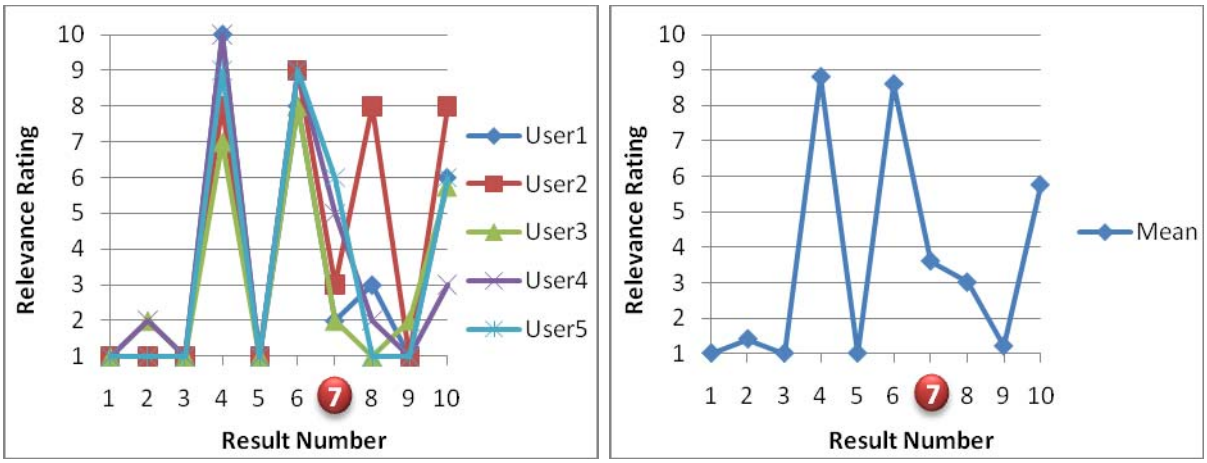


Figure 6-18 Query Seven Relevance Rating

Precision	0.30
Average Precision	0.29
Reciprocal Rank	0.25
Total Reciprocal Document Rank	0.52
Max Possible TRDR	2.93

Figure 6-19 Query Seven Performance Measures

6.2.5.2 Key Observations and Considerations

Query set A, which contained queries one to four, was designed to conduct searches for technical SQL content, related to the syntax of the programming language, which was within the scope of the web-crawl as defined by the educator through the keyword file. Figures 6-6 to 6-13 display the results associated with query set A. The aim of conducting these queries was to evaluate the performance of the OCCS in content cache generation. As the concepts in question are all covered by terms contained in the keyword file, relevant material should be available from the cache if the OCCS focused crawling is functioning as designed.

There was general correlation between assessors on the characteristics examined. The majority of the result graphs follow a similar pattern for all five domain experts. This gives confidence in the results returned. One of the most common causes of variation between participant's ratings tended to be when index pages were encountered[8]. Index pages essentially act as the table of contents for a web site. While some may contain relevant content, a page which could be reached from the index would be the most useful in addressing the information need of the query. One of the comments entered by an evaluation participant was: "This is an index page of what could be quite useful pages but in itself not useful". While some reviewers rated these pages as being of some relevance, others decided that they were not strictly relevant for the query.

A caching approach has been taken in the OCCS and a decision was made to only allow access to content within the cache, i.e. the hyperlinks on pages do not link back to the WWW. This can prove frustrating as relevant content is available but cannot be reached, particularly in the case of index pages. This strategy was intended to ensure that the learner could only access content within the educator-defined scope of the educational experience. There are two possible means of addressing this problem. Firstly, if only content in the cache is used for the initial search, then it may be feasible to allow the learner to link to further content on the WWW using the cache as an entry point. The second possible means of solving this issue

[8] The index pages which appeared in the results sets are denoted on each graph by a red circle around the relevant result number. For query one, results 5 and 7 were index pages. For query two, results 3, 6 and 8 were index pages. For query three, results 1 and 9 were index pages. The results set for query 4 contained no index pages. For query five, results 1, 2, 3 and 10 were index pages. For query 6, results 5 and 8 were index pages. Finally, for query 7, result 7 was an index page.

while maintaining the cache restriction is to remove the index pages from each site harvested. The index pages of a web site tend to be the first page encountered at the top-level of the site. Alternatively, Machine Learning techniques could be used to identify the structural pattern of index pages. The links from these pages could be extracted by the crawler but the content of the page not added to the cache. This would reduce the occurrence of index pages in results lists while still exploiting their link structure for content discovery.

The reasons why some of the results were deemed irrelevant can be explained upon examination. Due to the caching approach adopted by the OCCS which alters the link structure of pages, some results are missing embedded content which would have provided valuable material. Some pages also have interactive content, such as sample databases upon which it is possible to run test queries, which unfortunately, do not function when cached. These issues negatively impacted upon the ratings which the resources were assigned. Example participant comments in relation to these issues were: "Simple definition and example. Application window not working for interactive testing on a sample DB.", "Missing embedded links make this less useful" and "possibly would have been useful but many embedded items omitted".

There were a number of highly relevant, highly ranked resources for each query performed in query set A. This was an essential outcome, as it is an objective of this research to provide these caches of content for use in TEL scenarios through the U-CREATe interface. As a result, it was important that the focused crawling performed by the OCCS was caching content of high relevance for the scope of the crawl, and that this content was highly ranked for typical queries which learners would perform.

Figure 6-20 below, details the overall IR performance measures for the OCCS for query set A. The mean precision of the system was 0.7 which means that on average, seven of the top ten results were deemed relevant for the query performed. The MAP value for queries one to four is very positive at 0.95 from a maximum possible 1. This metric, as discussed above, not only addresses the number of relevant results returned, but also the relative ranking of these results. Although the number of queries conducted is quite small, this MAP value provides a positive indication of the performance of the OCCS in relation to the discovery and harvesting of content within the scope of the crawl, as defined by the keyword file.

Mean Precision	0.7
Mean Average Precision	0.95
Mean Reciprocal Rank	1.00

Figure 6-20 Overall Performance Measures – Queries 1-4

The first four queries also perform well in terms of reciprocal document ranking. The top five results in queries one and two are all relevant, the top four results are relevant in query three and the top six results are all relevant in query four. This is a very desirable result as in an ideal scenario, the most relevant result to a query for the intended audience would always be first in the list. This is extremely difficult to achieve as the intent behind each query and the target audience are not known to the search engine. However, in the case of the OCCS, the range of audience is much narrower and the queries more specific, as such the amount of possible ambiguity in the information need is reduced. The MRR measure for the first four queries was a maximum possible value of 1.

Despite the encouraging performance of the OCCS for these four queries, there are still some results which were deemed irrelevant, diluting the results lists for each of the first four queries. A total of 12 results from those examined for queries one to four were deemed irrelevant. Of these 12 results, three were index pages pointing to SQL content. Of the other nine irrelevant pages, only one was unrelated to the scope of the web crawl. This means that although these results were deemed irrelevant to the information need which produced the search query, the majority are resources which are within the scope of the web crawl and are validly contained in the content cache. This is an important distinction, if a large amount of content outside the scope of the crawl was appearing in the results lists, then the performance of the focused crawling functionality of the OCCS could be called into question. However, this result set dilution does highlight the need to further examine the performance of the indexing techniques and query-document similarity measures employed by the OCCS. These approaches need improvement in an effort to reduce the number of irrelevant resources appearing in the results lists.

Both the evaluation metric results and the overall performance of the OCCS deteriorated markedly for query set B, which contained queries five, six and seven. Figures 6-14 to 6-19 display the results associated with query set B. This deterioration in performance was to be expected as the concepts in question were not covered by terms in the keyword file. However, the degree to which performance was affected was quite surprising. In query five, "Schema", four of the results were indexes as these were the only pages where there was content related to the schema concept in a page which still passed the classification stage of the crawl. These pages were still deemed to be irrelevant by the participants and, in fact, none of the top ten results were deemed relevant for this query.

The performance of queries six, "Relational Databases", and seven, "What is a Stored Procedure", was slightly better, achieving precision ratings of 0.4 and 0.3 respectively. Query six even achieved a respectable average precision rating of 0.89 and a reciprocal rank of 1. This can be attributed to the fact that although there was no Relational Database term in the keyword file, there was a general SQL Database term which may have harvested some relevant terms which were included in the classification model. However, overall, the performance of these three queries was well below an acceptable level.

The performance of these three queries demonstrates the importance of keyword file generation in OCCS process flow and reinforces the fact that this should be an educator-driven exercise to define the scope of the subject area. These queries prove that the classification process is quite rigorous and that minimal amounts of peripheral, out-of-scope content is harvested during the crawl. The SQL ontology which was used for keyword generation in this evaluation has since been expanded to include conceptual elements, such as schema and stored procedures, along with the existing technical language syntax.

With regard to query seven, stop-word removal is also an issue. The performance of the query is significantly improved if the words "What is a" are removed. This is due to the query-document similarity measure employed by the search interface. It is a combination of the boolean model and vector space model. The index is searched for documents which contain all the terms of the search query. The identified documents are then ranked based upon a vector space model. NutchWAX employs an n-gram based solution for common terms, or stop-words. This is designed to improve the efficiency of searches. A list of

common terms is maintained and any items in this list are indexed with their direct neighbours at index time. For example consider the string “Alexander the great”. This would be indexed as four individual terms as “the” is identified as a common word: Alexander, Alexander-the, the-great and great. However, in the case of query seven, this does not improve question answering. “a” is the only term deemed common by the default NutchWAX implementation, therefore “What is a Stored Procedure” is broken down into six terms: What, is, is-a, a-Stored, Stored and Procedure. This means that the initial boolean filtering of the search is still looking for documents which contain all these terms and unless the string “What is a Stored Procedure” occurs in the document, it will struggle to identify relevant content.

This approach is beneficial for phrase searching as it maintains the relationship between neighbouring terms rather than simply removing common terms. The removal or conversion to n-grams of common terms is, at times, a balancing act. The classic example in general purpose search is an individual searching for the Shakespearian quote “to be or not to be”. Many of these terms may have been removed from the index as stop-words making the IR task much more difficult. However, in the case of the OCCS where the entire domain is known at index time, there could be a case for removing all but the ontology terms. If the ontology was fine-grained enough this would reduce each search down to the conceptual elements of the language. Further experiments on these approaches are required before a judgement on the best strategy can be made.

Mean Average Precision	0.71
Mean Reciprocal Rank	0.75

Figure 6-21 Overall Performance Measures – All Queries

The overall performance metrics of the OCCS for the seven queries conducted are detailed in figure 6-21 above. These values are down on those achieved by queries one to four as they are obviously affected by the poor performance of queries five, six and seven. The MRR implies that, on average, the first relevant result is between rank one or two in the results list. The MAP is a positive figure considering one of the queries received a MAP of zero which is reducing the mean quite considerably. While the number of queries conducted is relatively small, the fact that the content has been examined by domain experts and received generally

positive assessment values is an encouraging indication of the overall performance of the OCCS.

6.2.5.3 Content Quality and Technical Validity

The second and third criteria that each of the evaluation participants were asked to rate the results list for were technical accuracy and quality. Participants were informed that the content was to be used in an undergraduate class of students who had begun to learn about SQL but are relative beginners. They then examined the content to ensure that it was accurate and that there were no technical inconsistencies or irregularities. The participants also assessed if the content was of sufficient quality for this context.

An immediate piece of feedback from the evaluation participants was that questions two and three should only have been asked if the content was deemed relevant to the query by the value assigned to the first likert scale. In the words of one of the participants: "if I did a search for "sheep" and it returned a result about "cows" it would not be relevant to the query but could be factual, and an excellent site about cows". In other words, content could be of high quality and technically accurate but lie outside the scope of this analysis. Consequently, the domain experts may have been unqualified to judge such irrelevant content for these criteria. Therefore it was decided to only assess the quality and technical accuracy of content that was deemed relevant by the evaluation participants. The following set of figures display the values assigned for questions two and three, for each of the relevant results, for each query performed. Figures 6-22, 6-25, 6-28, 6-31, 6-34 and 6-37 are used to indicate the results deemed relevant for each of the queries performed. Figures 6-23, 6-26, 6-29, 6-32, 6-35 and 6-38 illustrate the results assigned to each relevant resource for technical validity. Figures 6-24, 6-27, 6-30, 6-33, 6-36 and 6-39 illustrate the results assigned to each relevant resource for content quality. There is also an associated figure for each query which displays the mean of these quality and technical accuracy ratings.

Query Set - A

Query 1: SQL Select Statement

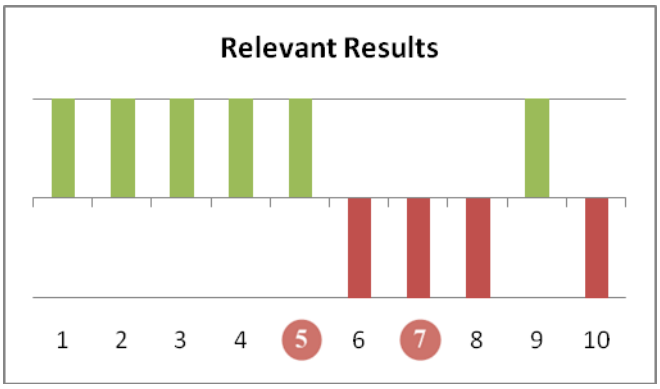


Figure 6-22 Query One - Relevant Results

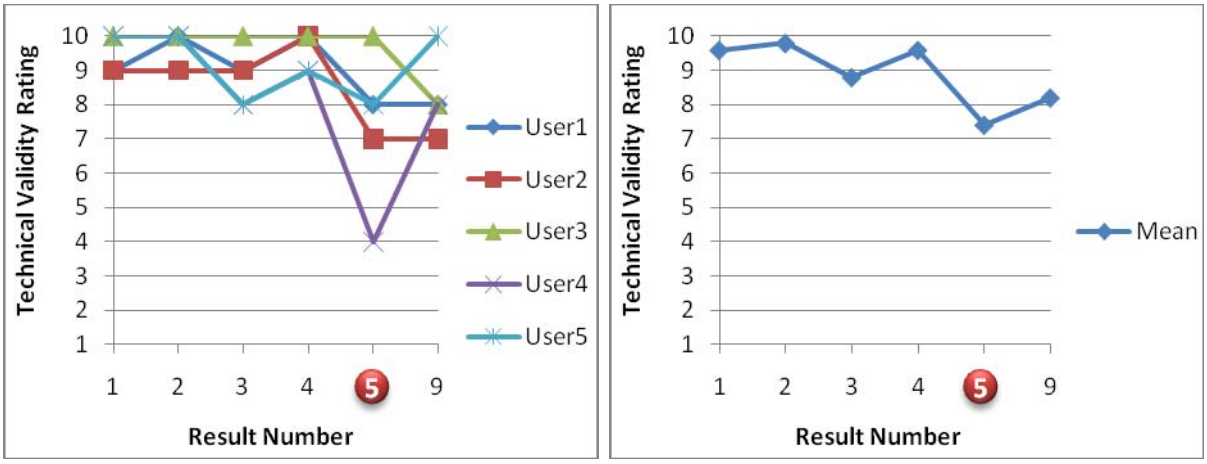


Figure 6-23 Query One – Technical Validity

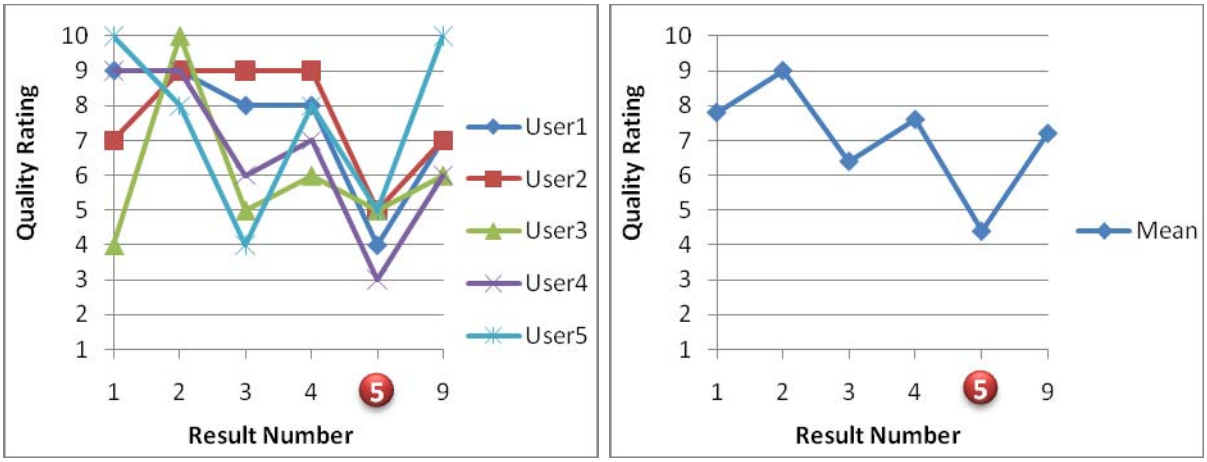


Figure 6-24 Query One – Content Quality

Query 2: SQL Insert Statement

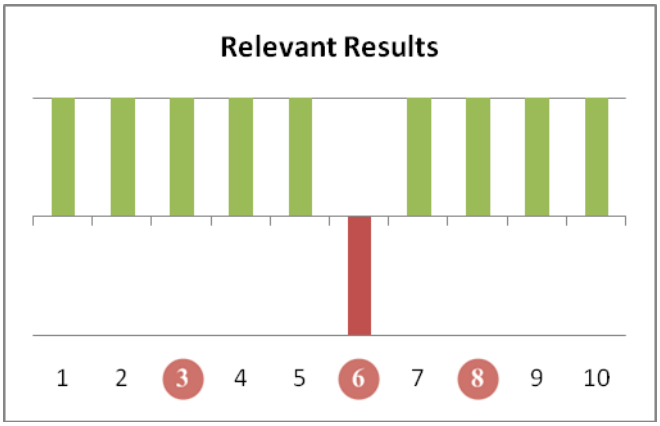


Figure 6-25 Query Two - Relevant Results

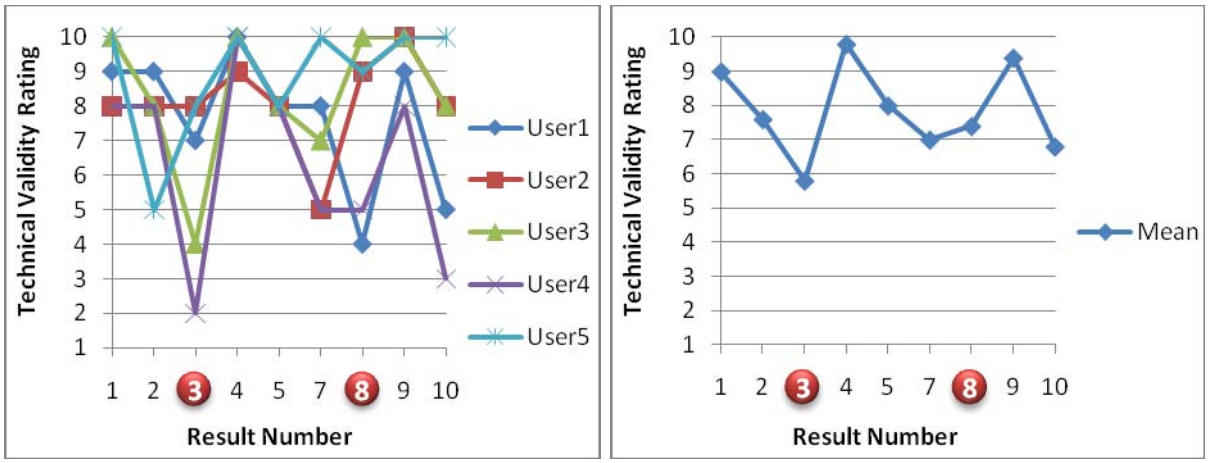


Figure 6-26 Query Two – Technical Validity

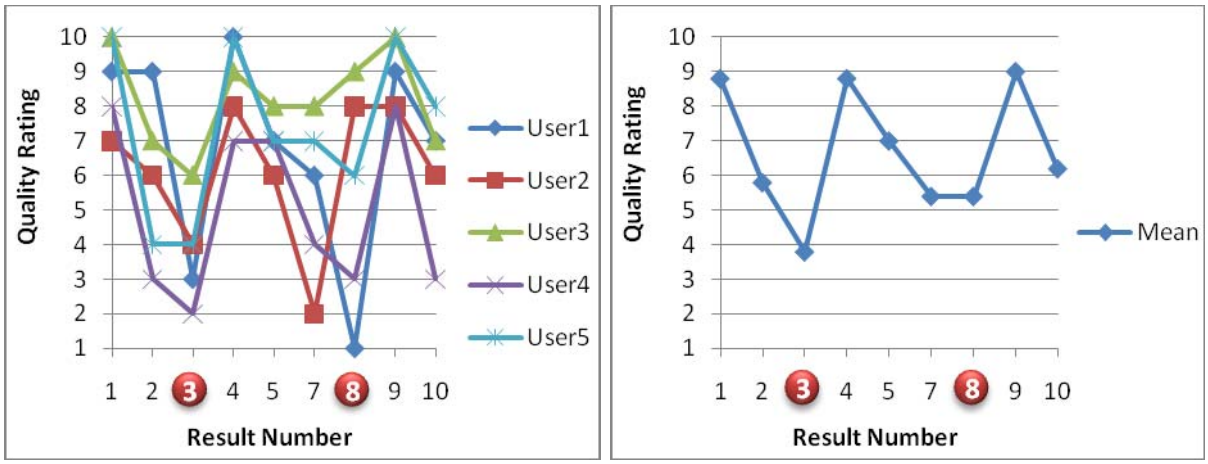


Figure 6-27 Query Two – Content Quality

Query 3: SQL “Create View”

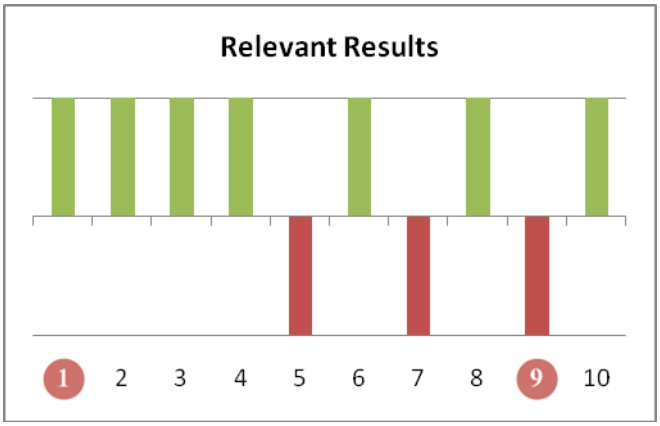


Figure 6-28 Query Three - Relevant Results

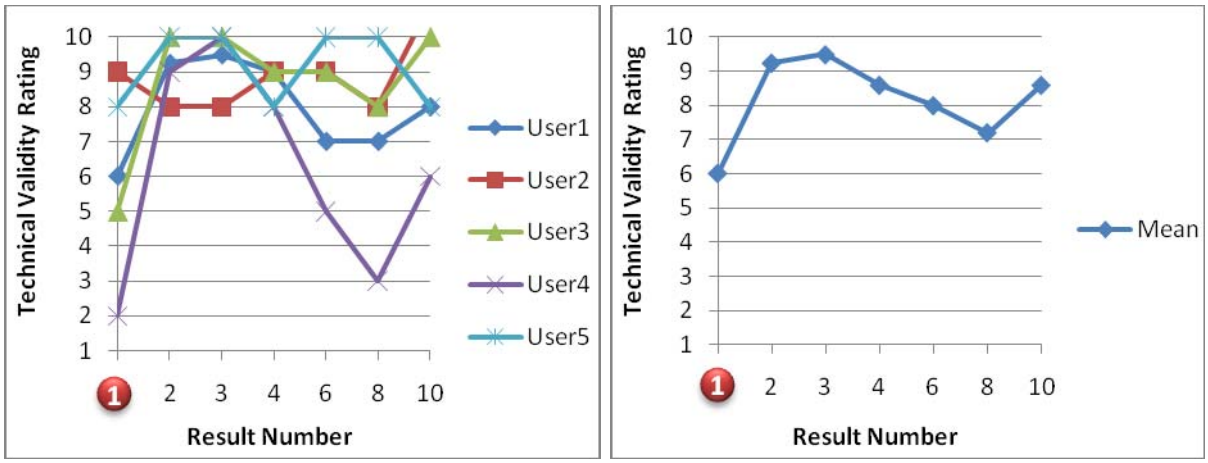


Figure 6-29 Query Three – Technical Validity

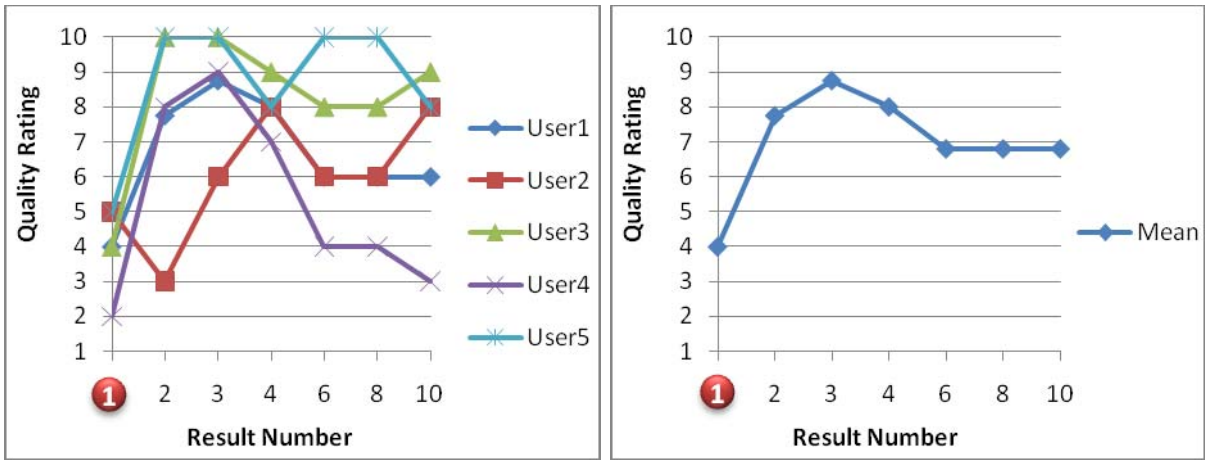


Figure 6-30 Query Three – Content Quality

Query 4: Foreign Key

Figure 6-31 Query Four - Relevant Results

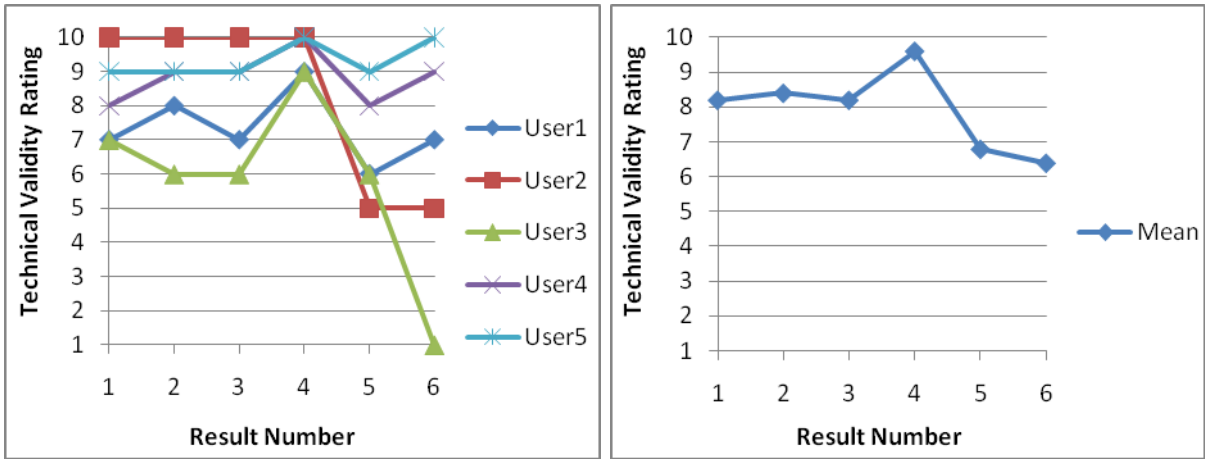


Figure 6-32 Query Four – Technical Validity

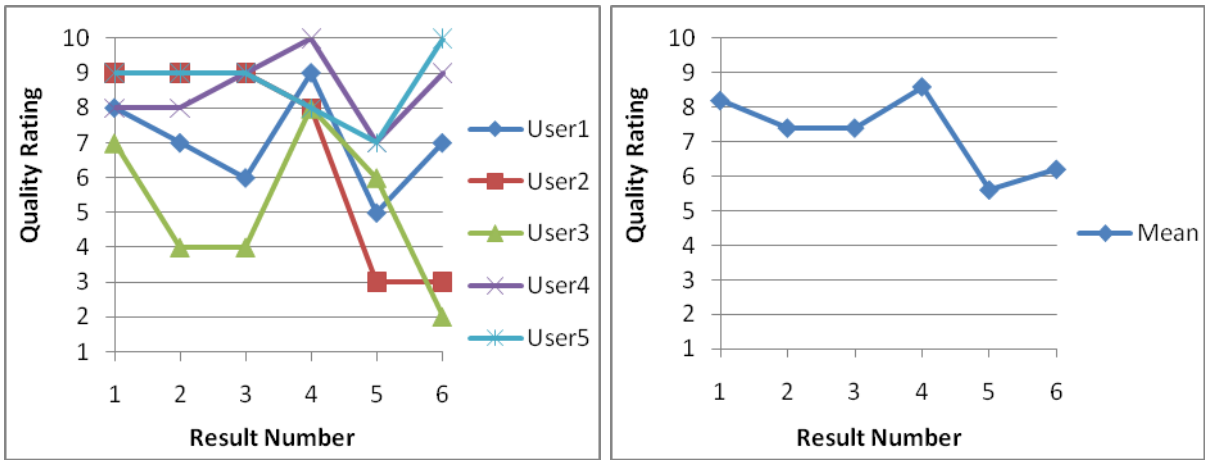


Figure 6-33 Query Four – Content Quality

Query Set – B

Query 5: Schema

As discussed above, none of the top ten results were deemed relevant for query five. As a result, there is no assessment of the results list for technical validity and quality.

Query 6: Relational Databases

Figure 6-34 Query Six - Relevant Results

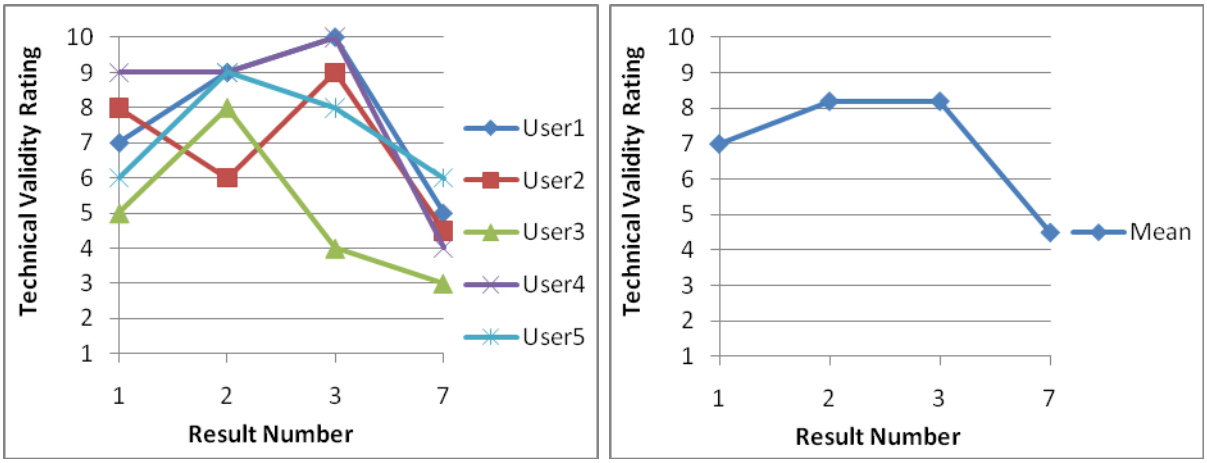


Figure 6-35 Query Six – Technical Validity

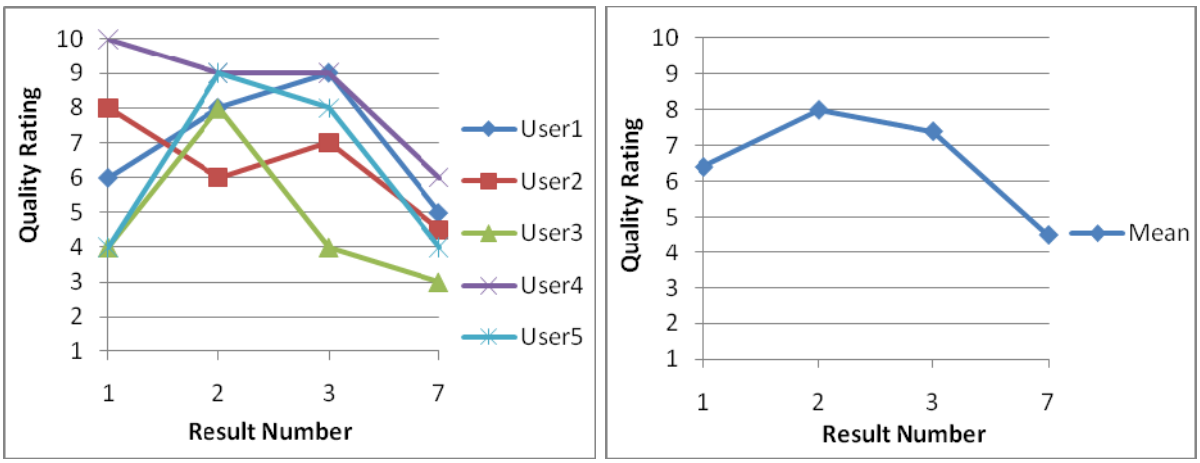


Figure 6-36 Query Six – Content Quality

Query 7: What is a Stored Procedure

Figure 6-37 Query Seven - Relevant Results

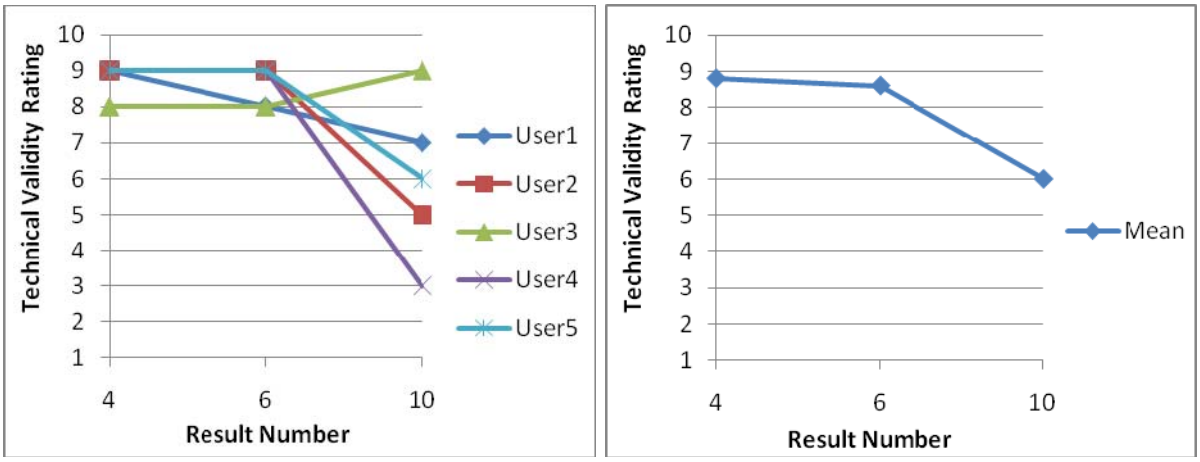


Figure 6-38 Query Seven – Technical Validity

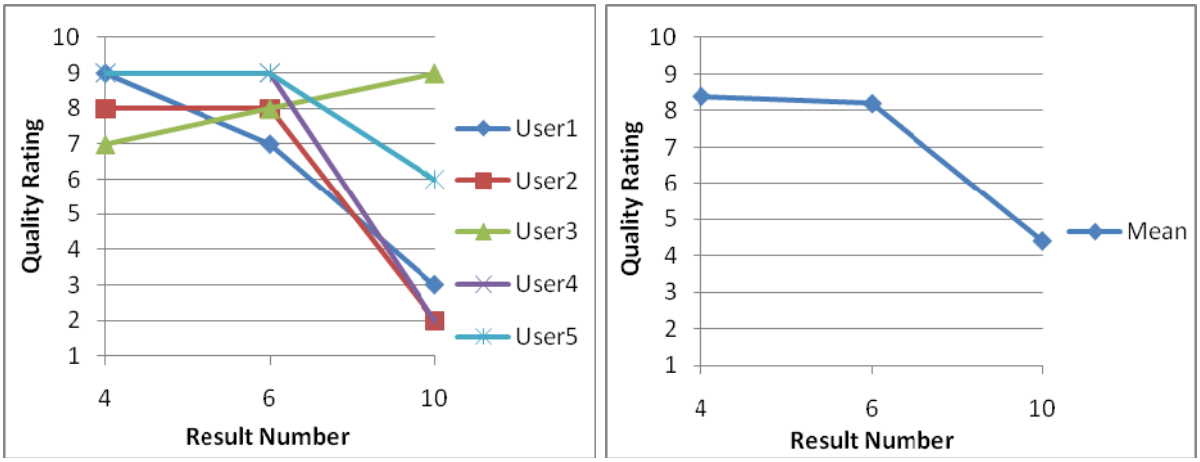


Figure 6-39 Query Seven – Content Quality

6.2.5.4 Key Observations and Considerations

There was considerably more variation between participants in their ratings of the more subjective questions two and three, technical validity and content quality. It is expected that some of the characteristics of content are more readily measurable than others. The domain expert can quite easily rate the relevance of a piece of content, as they are aware of the query performed, the information need behind the query and of the target audience for the query. However, the quality of each piece of content is more subjective, abstract and difficult to quantify.

Some of the variation in the quality and technical accuracy ratings assigned by the participants can be accounted for by the index problem discussed in the relevance section above. While rating an index page to be relevant to a particular query, some of the participants have decided that some of these pages are not high quality results as you cannot directly see the technical content related to the concept in question.

	Mean Technical Validity	Mean Quality
Query 1	8.90	7.07
Query 2	7.87	6.69
Query 3	8.16	6.99
Query 4	7.93	7.23
Query 5	N/A	N/A
Query 6	6.98	6.58
Query 7	7.80	7.00

Figure 6-40 Mean Technical Validity and Quality of Relevant Results

Despite the ratings for indexes reducing the overall values for technical accuracy and content quality, the mean values for these metrics, illustrated in Figure 6-40 above, are still encouraging. This was an important outcome in the context of this evaluation as it demonstrates that the content being returned for searches performed in the OCCS is technically correct and of sufficient quality to be valid for use in educational offerings.

6.2.6 Experiment One – Summary and Overall Observations

The analysis of both the focused crawls conducted by the OCCS and the search results delivered for each query performed on the OCCS content cache produced some interesting and valuable results. This experiment aimed to evaluate the performance of the OCCS in generating caches of content from open corpus sources which are within an educator defined scope and suitable for inclusion in an educational offering.

A focused crawl was conducted to discover and harvest content within the SQL subject domain. A crawl scope was defined in consultation with educators in the area and the classification training process completed with manual educator review of the outputs. The crawl ran for a total of 43 hours, generating a cache of 149,933 resources. The provision of inputs for the crawl and the manual review of the classification training outputs is currently quite crude and needs to be implemented in a more intuitive, user-friendly manner. The memory use of the crawler also needs to be examined. While it didn't cause a major problem during these crawls, it should be examined before future extension of the OCCS.

Once the crawl was completed, five educators in the domain of SQL with experience of TEL were asked to perform a manual review of the content cache for typical content searches. Each participant conducted seven searches and rated the top ten results for relevance, technical validity and quality. These seven searches were divided into two query sets. Query set A contained searches for technical SQL content within the scope of the crawl as defined by the educator. Query set B contained searches related to SQL but peripheral to the syntax of the language and outside the scope of the crawl.

There was a marked difference between the metric results for query sets A and B. The performance of the OCCS for query set A was encouraging with positive metric results for all four queries. However, the metric results for query set B were not acceptable if such content was sought for during an educational offering. This disparity highlights the importance of scope definition in the focused web crawl and reinforces the significance of exploiting the educator's knowledge and expertise. The authoring of the keyword list which drives both the classifier training and seed generation processes is crucial and methods of providing improved guidance and assistance during this process could prove valuable to the educator.

Further experimentation on this point is necessary when designing the user interface for educator authoring of inputs and review of training outputs.

The results of this experiment are generally positive with regard to the performance of the OCCS, however there are improvements, modifications and additions which can be made to the system to better address the objectives of this research. The handling of index pages when generating collections of content needs to be examined in more detail. These pages are valuable hubs of links to relevant resources, however when included in the cache they potentially dilute ranked results list with what is essentially uninformative content. Suggestions for handling such pages have been provided in 6.2.5.2. These include enabling the links on index pages to allow the learner to navigate back to the live web content or using URL depth and machine learning techniques to identify such pages and prevent their presentation in results lists. However, more research is necessary to identify the most beneficial course of action in this regard.

The caching approach adopted by the OCCS was a deliberate design feature as it was hypothesised that these content caches could be used as fuel, and the OCCS as a feeder service, for future content analysis and repurposing services within TEL applications. It was also felt that by providing controlled, subject-specific collections of content, the risk of content dilution and learner distraction encountered on the open WWW, would be minimised. However, this caching approach has caused some problems with the display of pages that contain embedded or interactive content in the OCCS. An examination of the link-handling implemented in the OCCS and experimentation with URL redirection will be conducted during the next OCCS development cycle.

The results presented in the above sections indicate that the OCCS focused crawling functionality is caching highly relevant, high quality content. The set of trial participants may be considered quite small and not statistically significant. However, it was necessary that the group of content assessors be restricted to educators with strong experience in the SQL domain. This ensured the accurate assessment of content relevance, quality and technical validity for the specific TEL scenario in which it was to be used. The results produced by these assessments provide indications of the performance of the OCCS in generating subject-

specific caches of content. Automated approaches to content cache relevance assessment will be trialled and evaluated in the future and are discussed in section 7.4, Future Work.

6.3 Experiment Two: Open Corpus Content Utilisation in TEL

6.3.1 Experiment Objective

A series of educational, technical and usability requirements for a prototype TEL application which enables the incorporation and resource-level reuse of open corpus content were specified in section 4.4 of this thesis. U-CREATe was implemented to meet these requirements as is detailed in section 5.3. A key objective of this thesis was to develop a demonstrator TEL application which could deliver a pedagogically beneficial, learner-driven educational experience using open corpus content. The educational requirements of this application were designed to ensure that the TEL experiences offered by U-CREATe are both engaging and pedagogically beneficial to the learner. As such it was necessary to evaluate U-CREATe in an educational scenario with real learners, whose performance could then be examined to determine if the learning offering delivered by the system was of benefit to the learner pedagogically.

U-CREATe is a TEL environment which is designed to be used as a supplement to learning conducted through traditional tutor-driven approaches. Control over the pace and direction of the learning offering is placed in the hands of the learner. The learner interacts with a mind mapping interface which is designed to improve learner engagement and promote reflection. The learner is provided with a cache of subject-specific content upon which to search for relevant information. These approaches reflect elements of Constructivism, Behaviourism and Enquiry-Based Learning. U-CREATe should facilitate the learner in refining their knowledge model of a subject area. A key element of Constructivism is that through reflection, the learner can build upon their existing knowledge models of the subject area.

The learner is the most significant stakeholder in TEL. Therefore, any approach to the use of open corpus content in TEL should produce educational offerings that are both beneficial for, and satisfactory to the learner. The objective of this experiment is to examine these two performance measures for an educational exercise conducted using the U-CREATe interface and OCCS content cache. By ascertaining if a student increased their knowledge of a concept through the completion of an educational offering in U-CREATe, it can be determined if the

interface combined with the use of open corpus content is beneficial to the learner. The sentiment of the learner towards the educational offering can be gauged by supplying each individual with a means of providing feedback on the experience upon completion of the exercise.

Several iterative trials of the U-CREATe system were conducted with various participant groups of students from Trinity College Dublin. These student groups included:

- A selection of postgraduate PhD students from the Knowledge and Data Engineering Group
- The Junior Sophister undergraduate students in the 3BA25 Information Management course in the BA (Mod) Computer Science Degree Program.
- The Junior Sophister undergraduate students in the CS3 Computer Science course in the BSc. Computer Science (Evening) Degree Program.
- The Junior Sophister undergraduate students in the ST3001 Computer Applications course in the Management Science and Information Systems Studies Degree Program.
- The Senior Sophister undergraduate students in the ST3001 Computer Applications course in the Engineering with Management Degree Program.

The trial results which are presented in the following section pertain to the most recently conducted of these evaluations.

6.3.2 Experiment Methodology

Each learner participating in the trial was provided with information detailing a simple relational database which they were required to create and relational schema diagram of this database, see figure 6-41. The learner was asked to create a mind map using U-CREATe and create a node in the map for each of the SQL statements necessary to create the database tables. A collection of data to be inserted into the tables was also provided, and the learner was asked to create a node for each of the SQL statements used to insert the data. Finally the learner was asked to query the database for defined pieces of information and create a node for each of the SQL statements which would be used. The learner was asked to write the complete SQL statement for each task into the mind map node. These nodes were to be grouped hierarchically based upon whether they were used to create, populate or query the database. The information provided to the participants and the entire list of exercise tasks can be found in Appendix F.

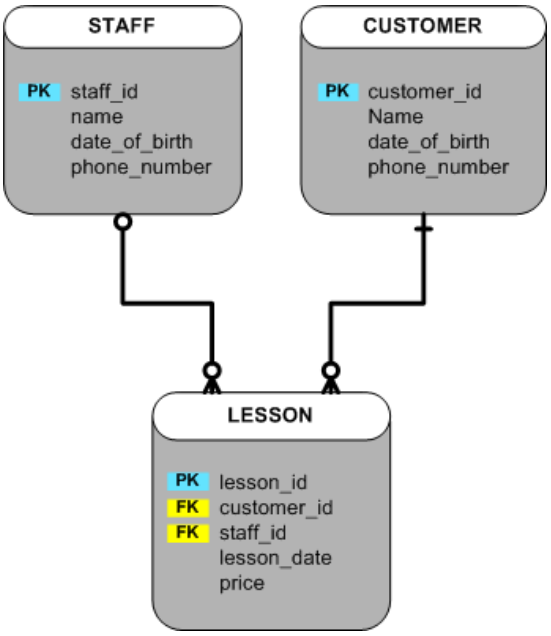


Figure 6-41 Main Trial Database Relational Schema Diagram

The OCCS was available to the learner at all times during the exercise. The participants were asked to use the OCCS to search for information to aid them in the composition of the required SQL statements and not to use other sources of information such as Google. If content was found in the OCCS which was particularly useful in writing an SQL statement, the learner was required to insert a link to that piece of content in the node in question.

6.3.3 Evaluation Metrics Employed

In order to assess if a learner has improved their knowledge and understanding of a concept through the completion of an educational experience, it is necessary to measure their competency with regard to that concept both before and after the educational experience. Therefore, before the students began the exercise in U-CREATe, they were all asked to complete a pre-test in isolation and without access to a computer. This pre-test posed questions related to the topics that were to be explored in the trial exercise itself. Once the students had completed the U-CREATe exercise, they were given a post-test. This test posed questions related to the same topics as both the pre-test and the exercise. These tests made it possible to measure the difference in the performance of each student between the pre and post tests as in figure 6-42. This gave some indication if the student had improved their knowledge and understanding of each concept by completing the exercise in U-CREATe. The full pre and post tests used in this evaluation can be found in Appendix F.

Pre-Trial Question 1	Post-Trial Question 1	Both Examine
Write the SQL statement required to create a database table called "EVENT". The table should contain the following fields event_code - A number between 1 and 9999 venue_code - A number between 1 and 9999 event_name - Text detailing the name of each event date - The date the event takes place	Write the SQL statement required to create a database table called "TICKET". The table should contain the following fields ticket_code – A number between 1 and 9999 event_code – A number between 1 and 9999 retailer – Text detailing the name of the ticket seller date_of_issue – The date the ticket was issued	CREATE TABLE Statement Basic Statement Structure Data Types
What would need to be added to the statement you created in part A to include the event_code field as the Primary Key of the table.	What would need to be added to the statement you created in part A to include the ticket_code field as the Primary Key of the table.	Primary Key Declaration
What would need to be added to the statement you created in part A to include the venue_code field as a Foreign Key of the table. The venue_code Foreign Key can be found in the table called VENUE.	What would need to be added to the statement you created in part A to include the event_code field as a Foreign Key of the table. The event_code Foreign Key can be found in a table called EVENT.	Foreign Key Declaration

Figure 6-42 Pre and Post Test Question Mapping

A learner's satisfaction and engagement with a learning experience can often be unrelated to the educational benefits of the experience. It can be the case that regardless of how beneficial a process is, the student can feel dissatisfied or frustrated by the experience. This can lead to problems with engagement and as a result, can have a negative effect on learning. It was felt

that the best way to measure the learners' satisfaction with the educational experience was to ask them to complete a usability questionnaire once the exercise was complete.

There are no absolute measures of system usability across all domains, as the usability of a system can only be defined with reference to the context in which the system is used. Despite this fact, broad general measures of usability which can be compared across a range of contexts are sometimes necessary. The System Usability Scale (SUS) is a reliable, low-cost usability scoring system that can be used to make general assessments with regard to a system's usability. SUS was developed by Digital Equipment Corporation in 1986 [Brooke 96] and is a questionnaire composed of ten statements relating to individual aspects of the system. The user specifies their level of agreement to the statement using a five-point Likert scale [Likert 32]. Despite its relative simplicity, SUS has been demonstrated to yield reliable results across a variety of sample sizes when compared to other usability assessment questionnaires [Tullis & Stetson 04].

As usability questionnaires are very quick to complete and require little effort on the part of the user, it was decided to use two separate questionnaires. One SUS questionnaire to measure the general usability of the system and a second, context specific questionnaire to measure the performance of the system with regard to the educational scenario in which it was employed. This questionnaire asked direct questions with relation to specific aspects of U-CREATe and the OCCS, and also allowed the learner to add any comments that they wished. The questions that the learner was asked were as follows.

How useful did you find this as an educational experience?

Do you believe this process would be beneficial in supporting the learning of a subject area?

How easy did you find the U-CREATe interface to use?

Are there any improvements that you would suggest making to the U-CREATe interface?

Was it difficult to find relevant content in the OCCS?

Was the OCCS interface intuitive to use?

Are there any improvements that you would suggest making to the OCCS interface?

Any other comments?

The full detail of the both the SUS and context-specific questionnaires used in this evaluation can be found in Appendix F.

6.3.4 U-CREATe Trial Results

The most recent trial of U-CREATe was conducted with undergraduate students from the ST3001 course in Software Applications in Trinity College Dublin. This class consisted of 28 students in both Junior Sophister and Senior Sophister years of study. The class were all beginners in the subject area, they had completed three tutorials on the basics of SQL prior to conducting this trial. None of the students had previous SQL experience before these tutorials. The trial was conducted in a one hour lab session and the students were presented with an introduction to U-CREATe before beginning the exercise. Once the trial was completed, all the pre-tests, post-tests and exercise answers were graded to ascertain each student's performance.

As discussed in the previous section, the metric which is most indicative of system performance for the purposes of this evaluation is the level of knowledge gain displayed by each student between their pre-test and post-test. Knowledge gain was calculated by assigning a percentage grade for each conceptual component of each question. The questions in the pre and post-tests examine the same concepts and as such the difference between the grades achieved for each of these concepts in the pre and post test can be compared. This gave the amount that the grade had improved, or deteriorated in rare cases, for that particular portion of the question between the pre and post tests. Figures 6-43, 6-44, 6-45 and 6-46 below present these values per student, for each question.

Question 1

% Measureable Knowledge Gain

	Part i									Part ii			Part iii			Overall
	Statement			Structure			Datatypes			Primary Key			Foreign Key			
Student	Pre	Post	Gain	Pre	Post	Gain	Pre	Post	Gain	Pre	Post	Gain	Pre	Post	Gain	
1	100	100	0	90	100	10	50	100	50	0	100	100	0	25	25	37
2	100	100	0	100	100	0	100	100	0	100	100	0	25	25	0	0
3	100	100	0	80	100	20	75	100	25	0	100	100	0	25	25	34
4	100	100	0	80	100	20	25	75	50	0	100	100	0	100	100	54
5	100	100	0	50	80	30	15	75	60	25	100	75	0	100	100	53
6	100	100	0	100	100	0	100	100	0	90	90	0	60	60	0	0
7	100	100	0	100	100	0	75	75	0	100	100	0	25	25	0	0
8	50	100	50	0	90	90	25	0	-25	0	75	75	0	0	0	38
9	100	100	0	90	100	10	80	100	20	90	100	10	90	100	10	10
10	100	100	0	100	100	0	100	100	0	100	100	0	75	75	0	0
11	100	100	0	100	100	0	25	90	65	100	100	0	50	80	30	19
12	100	100	0	75	95	20	50	66	16	100	100	0	25	100	75	22.2
13	100	100	0	100	90	-10	100	100	0	100	100	0	100	100	0	-2
14	80	100	20	60	100	40	40	100	60	75	100	25	40	75	35	36
15	100	100	0	75	75	0	75	90	15	100	100	0	90	100	10	5
16	100	100	0	25	100	75	100	100	0	100	100	0	25	100	75	30
17	100	100	0	0	90	90	0	100	100	0	100	100	0	90	90	76
18	80	100	20	25	100	75	25	100	75	100	100	0	25	80	55	45
19	100	100	0	100	100	0	75	100	25	100	100	0	66	100	34	11.8
20	66	66	0	50	80	30	40	66	26	100	100	0	25	25	0	11.2
21	100	100	0	100	100	0	100	100	0	100	100	0	75	100	25	5
22	100	100	0	90	100	10	100	100	0	50	100	50	50	50	0	12
23	80	100	20	80	100	20	100	100	0	80	100	20	0	90	90	30
24	66	100	34	75	75	0	75	75	0	100	100	0	25	100	75	21.8
25	100	66	-34	75	80	5	25	90	65	100	100	0	25	80	55	18.2
26	0	100	100	0	100	100	0	75	75	0	100	100	0	90	90	93
27	100	100	0	100	100	0	75	75	0	90	100	10	25	100	75	17
28	100	100	0	100	100	0	100	100	0	100	100	0	100	100	0	0
Mean Knowledge Gain																
			7.5			22.7			25.1			27.3			38.4	24.2

Figure 6-43 Knowledge Gain Per-Student and Averaged – Question 1

Question 2 % Measureable Knowledge Gain																
	Part i									Part ii			Part iii			Overall
	Statement			Structure			Data Population			Nested Select			Max Value			
Student	Pre	Post	Gain	Pre	Post	Gain	Pre	Post	Gain	Pre	Post	Gain	Pre	Post	Gain	
1	75	100	25	100	100	0	75	75	0	0	0	0	0	0	0	5
2	75	75	0	50	50	0	50	50	0	0	0	0	50	0	-50	-10
3	0	100	100	0	100	100	0	90	90	0	25	25	0	50	50	73
4	25	100	75	50	100	50	50	75	25	0	0	0	0	0	0	30
5	50	100	50	50	100	50	66	66	0	0	0	0	0	0	0	20
6	100	100	0	100	100	0	90	90	0	0	0	0	50	50	0	0
7	50	75	25	0	100	100	0	75	75	0	0	0	0	0	0	40
8	0	100	100	0	100	100	0	66	66	0	0	0	0	0	0	53.2
9	66	100	34	25	100	75	25	66	41	0	0	0	0	0	0	30
10	75	100	25	90	100	10	80	90	10	0	0	0	0	0	0	9
11	90	75	-15	100	100	0	100	100	0	0	0	0	0	0	0	-3
12	75	100	25	25	100	75	50	90	40	0	0	0	0	0	0	28
13	0	100	100	0	100	100	0	75	75	0	0	0	0	0	0	55
14	0	75	75	0	25	25	25	25	0	0	0	0	0	0	0	20
15	0	100	100	0	100	100	0	75	75	0	0	0	0	0	0	55
16	75	100	25	25	100	75	50	90	40	0	0	0	0	50	50	38
17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
18	50	100	50	0	100	100	50	90	40	25	33	8	0	50	50	49.6
19	75	100	25	100	100	0	100	90	-10	0	33	33	50	50	0	9.6
20	50	100	50	25	50	25	50	90	40	0	15	15	0	0	0	26
21	100	100	0	100	100	0	75	90	15	0	0	0	75	50	-25	-2
22	100	100	0	90	100	10	90	90	0	0	33	33	0	0	0	8.6
23	50	100	50	0	90	90	0	70	70	0	0	0	0	0	0	42
24	0	100	100	0	100	100	0	90	90	0	33	33	0	0	0	64.6
25	75	100	25	0	100	100	0	90	90	0	0	0	0	0	0	43
26	0	50	50	0	0	0	0	25	25	0	0	0	0	0	0	15
27	75	100	25	25	100	75	40	75	35	0	0	0	0	0	0	27
28	50	100	50	25	100	75	25	75	50	0	0	0	0	0	0	35
Mean Knowledge Gain																
			41.8			51.3			35.1			5.3			2.7	27.2

Figure 6-44 Knowledge Gain Per-Student and Averaged – Question 2

Question 3

% Measureable Knowledge Gain

	Part i											
	Statement			Structure			* and Where			< Function		
Student	Pre	Post	Gain	Pre	Post	Gain	Pre	Post	Gain	Pre	Post	Gain
1	100	100	0	100	100	0	100	100	0	100	100	0
2	100	100	0	100	100	0	100	100	0	100	100	0
3	0	100	100	0	100	100	0	100	100	0	100	100
4	100	0	-100	75	0	-75	50	0	-50	100	0	-100
5	100	100	0	50	100	50	50	100	50	100	100	0
6	100	100	0	100	100	0	100	100	0	100	100	0
7	100	100	0	50	100	50	66	75	9	100	100	0
8	0	100	100	0	80	80	0	100	100	0	100	100
9	100	100	0	100	100	0	100	100	0	100	100	0
10	0	100	100	0	90	90	0	66	66	0	100	100
11	100	100	0	100	100	0	100	100	0	100	100	0
12	100	100	0	100	100	0	100	100	0	100	100	0
13	100	100	0	100	100	0	100	90	-10	100	100	0
14	0	100	100	0	100	100	0	100	100	0	100	100
15	100	100	0	100	90	-10	90	100	10	100	100	0
16	0	0	0	0	0	0	0	0	0	0	0	0
17	0	0	0	0	0	0	0	0	0	0	0	0
18	0	100	100	0	100	100	0	100	100	0	100	100
19	100	100	0	100	100	0	100	100	0	100	100	0
20	100	100	0	100	100	0	100	100	0	100	100	0
21	100	100	0	75	75	0	100	100	0	100	100	0
22	0	100	100	0	100	100	0	66	66	0	100	100
23	100	75	-25	90	100	10	100	100	0	100	100	0
24	25	100	75	25	100	75	25	66	41	100	100	0
25	100	100	0	90	100	10	100	100	0	100	100	0
26	0	0	0	0	0	0	0	0	0	0	0	0
27	75	100	25	100	100	0	100	100	0	100	100	0
28	100	100	0	100	100	0	66	66	0	100	100	0

Mean Knowledge Gain

		20.5		24.3		20.8		17.9

Figure 6-45 Knowledge Gain Per-Student and Averaged – Part i - Question 3

Question 3

% Measureable Knowledge Gain

	Part ii						Part iii						Overall
	Single Field			Nested / Join			Multiple Field			Date Handling			
Student	Pre	Post	Gain	Pre	Post	Gain	Pre	Post	Gain	Pre	Post	Gain	
1	80	80	0	0	0	0	80	80	0	25	25	0	0.0
2	100	100	0	0	0	0	100	100	0	0	0	0	0.0
3	0	100	100	0	0	0	0	100	100	0	0	0	75.0
4	0	0	0	0	0	0	0	0	0	0	0	0	-40.6
5	0	0	0	0	0	0	0	0	0	0	0	0	12.5
6	100	100	0	0	0	0	100	100	0	0	90	90	11.3
7	100	100	0	0	0	0	100	100	0	0	0	0	7.4
8	0	75	75	0	0	0	0	75	75	0	0	0	66.3
9	100	100	0	0	0	0	0	90	90	0	0	0	11.3
10	0	100	100	0	0	0	0	0	0	0	0	0	57.0
11	80	80	0	0	0	0	80	80	0	0	0	0	0.0
12	0	100	100	0	0	0	100	100	0	0	0	0	12.5
13	0	100	100	0	0	0	0	80	80	0	25	25	24.4
14	0	80	80	0	0	0	0	0	0	0	40	40	65.0
15	80	0	-80	0	0	0	50	0	-50	0	0	0	-16.3
16	0	0	0	0	0	0	0	0	0	0	0	0	0.0
17	0	0	0	0	0	0	0	0	0	0	0	0	0.0
18	0	80	80	0	0	0	0	80	80	0	80	80	80.0
19	0	0	0	0	0	0	0	100	100	0	0	0	12.5
20	80	80	0	0	0	0	80	80	0	0	90	90	11.3
21	25	25	0	0	0	0	25	0	-25	0	0	0	-3.1
22	0	80	80	0	0	0	0	80	80	0	33	33	69.9
23	50	50	0	0	0	0	0	50	50	0	50	50	10.6
24	25	100	75	0	0	0	0	0	0	0	0	0	33.3
25	0	100	100	0	0	0	0	100	100	0	0	0	26.3
26	0	0	0	0	0	0	0	0	0	0	0	0	0.0
27	0	100	100	0	0	0	0	100	100	0	0	0	28.1
28	100	0	-100	0	0	0	75	0	-75	0	0	0	-21.9
Mean Knowledge Gain													
			28.9			0.0			25.2			14.6	19.0

Figure 6-46 Knowledge Gain Per-Student and Averaged – Parts ii and iii - Question 3

The average knowledge gain for each concept assessed within the three questions varies quite considerably. To get a better impression of these variations, figures 6-47, 6-48 and 6-49 below, display the mean student performance in both pre-test and post-test for each concept examined. The variation in knowledge gain is clearly visible in these graphs.

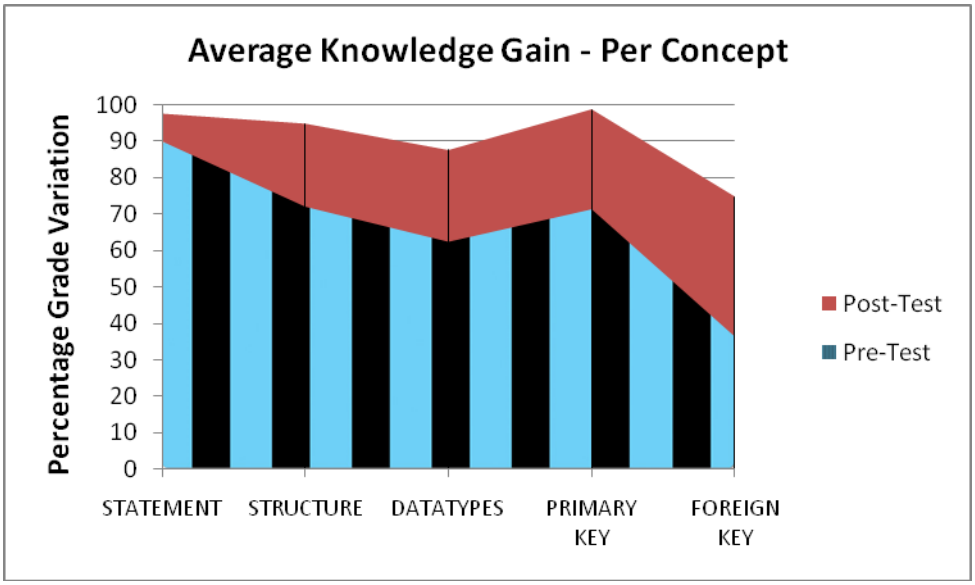


Figure 6-47 Mean Student Performance Per-Concept – Question 1

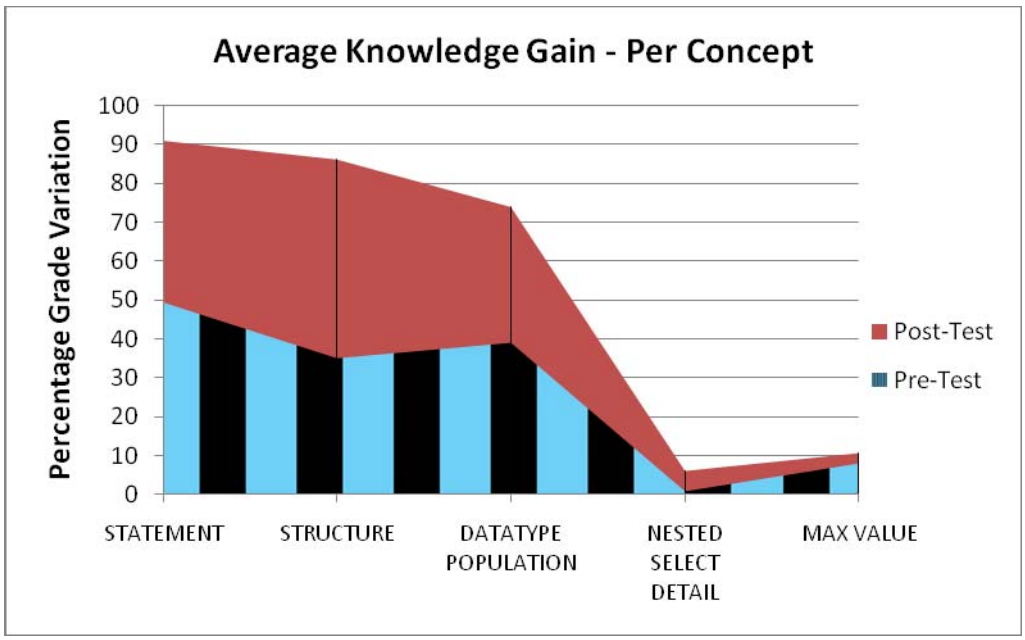


Figure 6-48 Mean Student Performance Per-Concept – Question 2

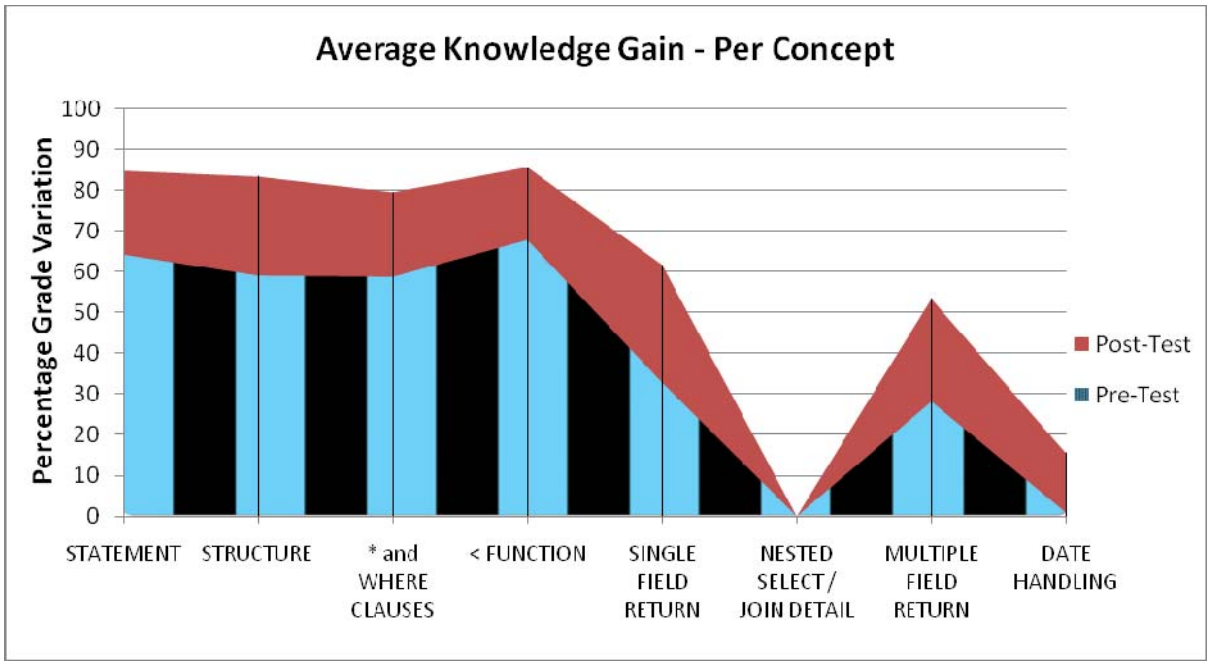


Figure 6-49 Mean Student Performance Per-Concept – Question 3

Figures 6-50, 6-51 and 6-52 below, show the distribution of knowledge gain levels amongst the students for each question. These graphs help to visually identify the average performance of students in the assessment and also identify individuals who performed exceptionally well or exceptionally poorly in the context of the class group.

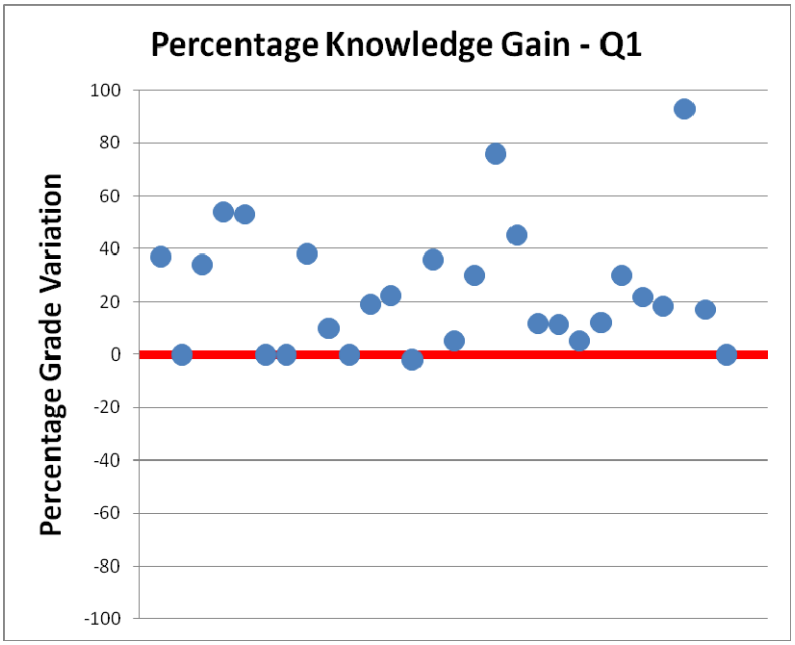


Figure 6-50 Knowledge Gain Spread – Question 1

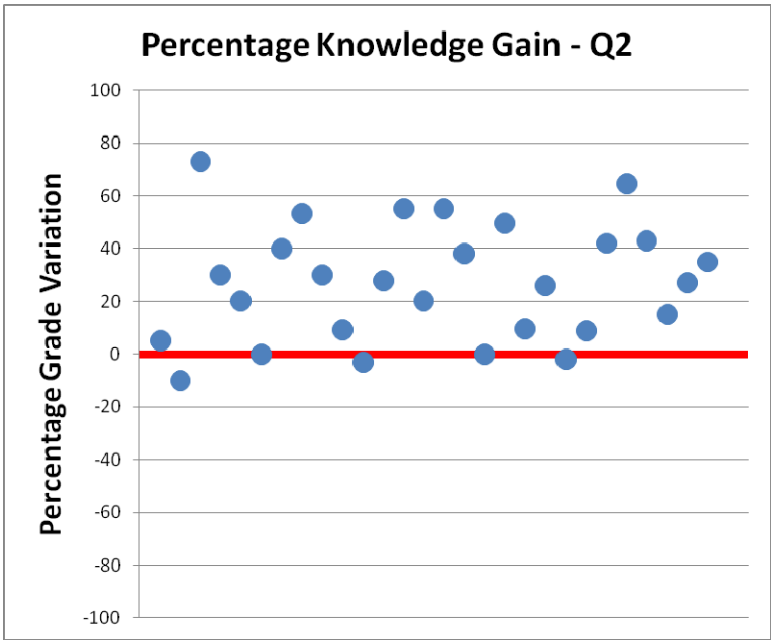


Figure 6-51 Knowledge Gain Spread – Question 2

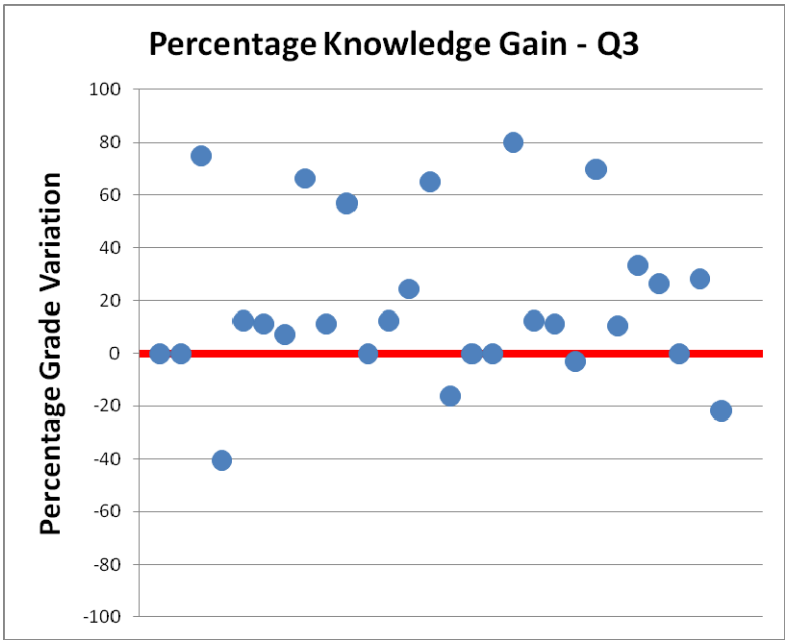


Figure 6-52 Knowledge Gain Spread – Question 3

6.3.4.1 Key Observations and Considerations

The results emerging from the examination of student performance in the trial were encouraging. 22 of the 28 student participants displayed positive knowledge gain for question one, 23 for question two and 18 for question three. This improvement in performance between the pre and post-tests demonstrates that the learning experience conducted using U-

CREATe had a positive effect on the majority of students knowledge of each of the concepts involved.

The students who participated in the trial detailed in the sections above had already completed some basic tutorials on SQL. They had begun to learn the basics of creating, populating and querying a database using SQL statements, however they had not yet covered any of the more advanced topics such as statement nesting, joins, date handling or functions. Figures 6-47, 6-48 and 6-49 clearly demonstrate that the improvement in performance was greatest for the concepts which has already been covered in previous tutorials. The concept of nested select statements and the MAX function, both examined in question two had not been covered in previous tutorials, nor had joins or date handling which are both examined in question three. The students performed particularly poorly in these parts of the questions. These findings reflect the educational design of U-CREATe, detailed in section 4.4.1, as an educational tool which should be used to supplement learning conducted through traditional tutor-driven scenarios. Students are utilising their existing knowledge models and through the process of reflection and exploration, are extending and formalising these models. U-CREATe would not effectively support the learning of a topic in isolation.

Figures 6-50, 6-51 and 6-52 demonstrate the spread of knowledge gain among the students. 79% of students displayed positive knowledge gain in question one, 82% in question two and 64% in question three. Some students displayed no measurable knowledge gain over the course of the trial: 17% in question one, 7% in question two and 21% in question three. Despite the fact that the overall level of knowledge gain in the student body is quite positive, the numbers showing no measurable knowledge gain is still significant. Examination and enhancement of the U-CREATe interface with the aid of a learning design expert could improve the performance of the application for more advanced concepts. Adaptive navigation and display in future developments of U-CREATe could be used to personalise the system for each learner, which could help to improve its performance for all the students.

One student performed quite poorly in question two. They showed equal performance in four of the five concepts examined, however they deteriorated in the final concept. This was the MAX function part of the question which they hadn't covered previously. While the student made an attempt at the question for which they received some marks in the pre-test, they did

not attempt the section of the question in the post-test. Three students performed noticeably poorly in question three. One of these students did not attempt question three at all in the post-test, despite performing reasonably on the same concepts in the pre-test. This may indicate that the student ran out of time. The other two students performed identically over the initial concepts in the question but did not attempt the more advanced concepts, despite attempting them in the pre-test, again this may indicate that they ran out of time. In the system usability questionnaire, where there was space for the students to make general comments, a number stated that they had run out of time during the exercise.

6.3.5 Usability Results

When processed, SUS questionnaires produce a single number, in the range 0 – 100, which represents a measure of the overall usability of the system being evaluated. This is a combined score for all the components and features of the system. SUS scores for individual items within the questionnaire are not considered meaningful in isolation.

The overall SUS score is calculated by first summing the score contributions from each item in the questionnaire for each user. The score contribution for each item is an integer in the range 0 to 4. For questionnaire items 1,3,5,7,and 9 the score contribution is the scale position assigned by the user minus 1. For items 2,4,6,8 and 10, the contribution is 5 minus the scale position assigned by the user. The sum of each of these score contributions is then multiplied by 2.5 to obtain the SUS value of system usability for that user. It is then possible to average these scores across all users to provide an overall SUS score for the system. Figures 6-53 and 6-54 below, present the distribution of SUS scores amongst the students in this evaluation.

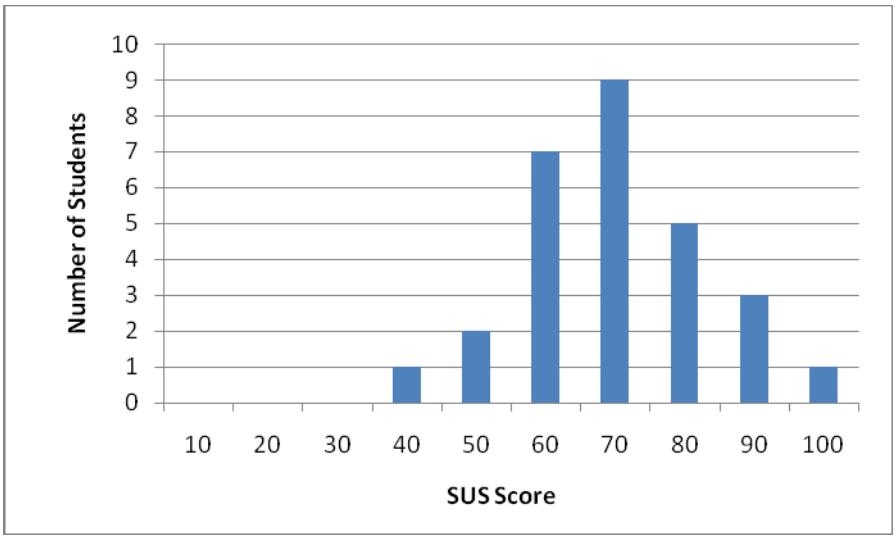


Figure 6-53 SUS Usability Score Distribution

Very Positive	91 - 100	1
	81 - 90	3
	71 - 80	5
Positive	61 - 70	9
Mildly Positive	51 - 60	7
Mildly Negative	41 - 50	2
Negative	31 - 40	1
Very Negative	21 - 30	0
	11 - 20	0
	0 - 10	0

Figure 6-54 SUS Usability Score Distribution

The context-specific usability questionnaire was not used to produce a single composite usability value for the system, but to gauge the students opinion of, and sentiment toward the various components and aims of the system. Figure 6-55 below, details the average response among the student group to each of the questionnaire items.

Question		Mean	Median
How useful did you find this as an educational experience?	Not at all Useful ... Very Useful 1 ... 10	6.89	7
Do you believe this process would be beneficial in supporting the learning of a subject area?	Not Beneficial ... Very Beneficial 1 ... 10	6.82	7
How easy do you find the U-CREATe interface to use?	Very Difficult ... Extremely Easy 1 ... 10	7.5	7
Was it difficult to find relevant content on the OCCS?	Very Difficult ... Extremely Easy 1 ... 10	6.63	6.5
Was the OCCS interface difficult to use?	Very Difficult ... Extremely Easy 1 ... 10	6.75	7

Figure 6-55 Usability Questionnaire Score Distribution. Scores out of 10.

The following graphs, figures 6-56, 6-57, 6-58, 6-59 and 6-60, show the distribution of responses from the students to each of the context-specific usability questionnaire items.

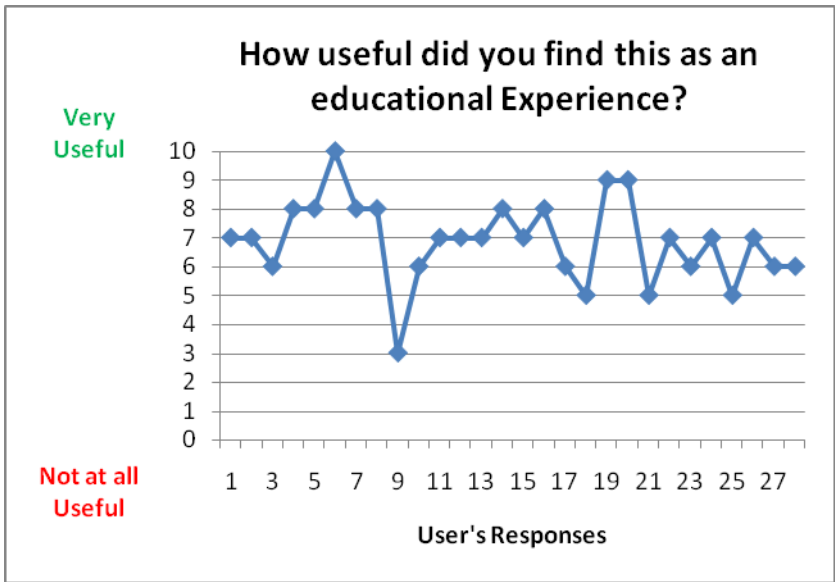


Figure 6-56 Usability Question 1

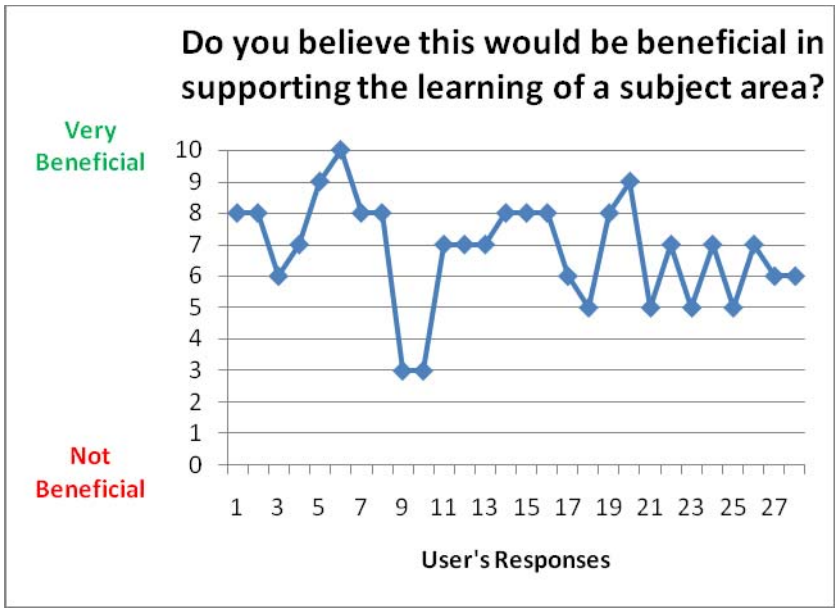


Figure 6-57 Usability Question 2

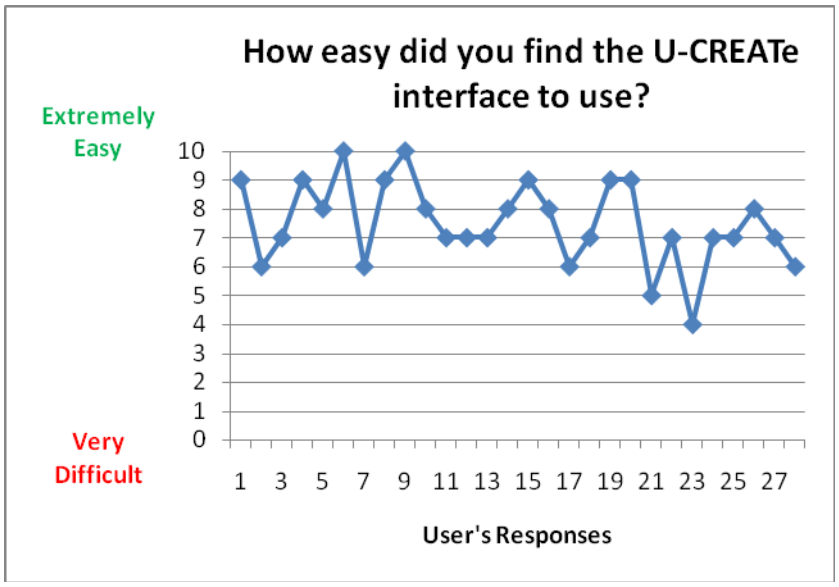


Figure 6-58 Usability Question 3

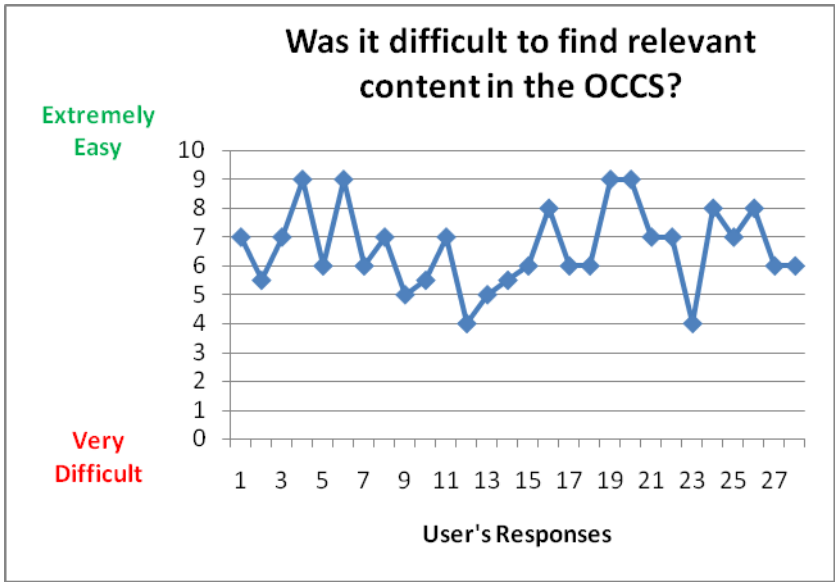


Figure 6-59 Usability Question 4

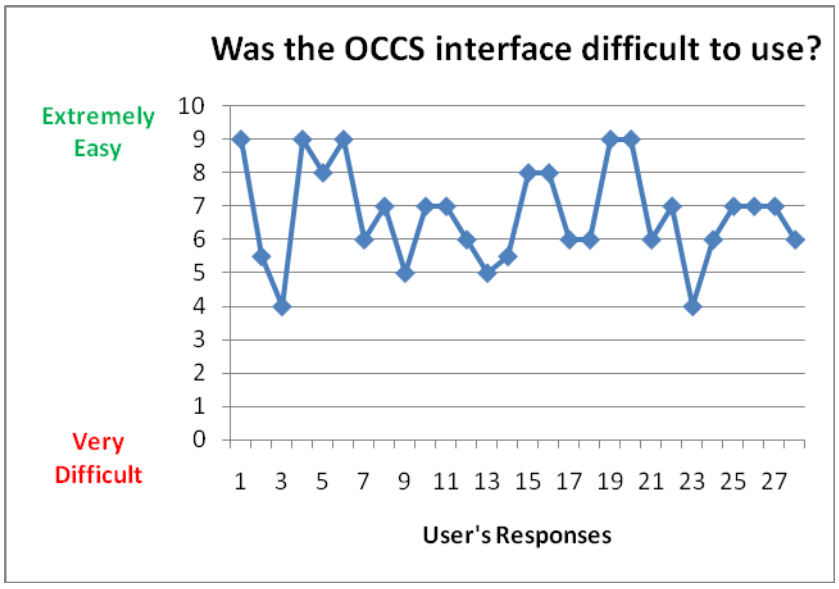


Figure 6-60 Usability Question 5

6.3.5.1 Key Observations and Considerations

As demonstrated by figures 6-53 and 6-54, the average overall SUS score for U-CREATe which was assigned by the students in this evaluation was 65.89. The median SUS score was 65, so positive or negative outliers did not significantly impact the average usability score. 25 of the 28 students produced a SUS score of above 50, which is a positive score. Of these, 9 of the 28 students rated the system above 70 which is a very positive usability score.

Figure 6-55 shows the mean and median responses to the questions posed in the context specific usability questionnaire. The responses to these questions were encouraging, all averaging between 6.5 and 7.5 out of a possible 10. This indicated that the students overall attitude, not only to the functions of the system, but to the type of educational experience offered by the system was positive in nature.

Figures 6-56 to 6-60 demonstrate that while there were rare negative outliers, the majority of the students displayed a positive attitude to the five context-specific usability questions. They felt that the educational experience offered by U-CREATe was useful and would be beneficial in aiding the learning of a subject area. The students generally found both interfaces intuitive to use. The students also found it relatively easy to find relevant content using the OCCS.

The students were provided with the opportunity to provide comments in relation to either system or the exercise as a whole, but few opted to enter such comments. Of the few who did, most were related to running out of time during the exercise. Examples include: "A good learning experience. The links are useful. Might be better having more time to check out what it can do"; "It was hard to get my head around at first but grand now"; "Could do with a bit less to do for the hour that we had". Future trials should reflect these concerns by providing less tasks or more time in which to complete the tasks.

6.3.6 Experiment Two – Summary and Overall Observations

There are a number of key findings that arose from this evaluation of U-CREATe. The evaluation was designed to measure both the effectiveness of educational offerings delivered by the system and the usability of the system. These key findings and considerations are summarised below.

The mean level of knowledge gain for the majority of students was positive and encouraging. There were some students whose performance produced negative results, however these were not indicative of the average student performance and can most likely be explained by the student in question running out of time. However, the performance of U-CREATe should be examined for all students and these negative results should not be ignored. Further iterative design and experimentation will be conducted in an attempt to improve the overall performance of the application.

The more advanced concepts examined displayed a lower average knowledge gain than the basics which the students had already begun to learn in previous tutorials. This indicates that the current design of U-CREATe is most beneficial and effective as a tool for student reflection and knowledge reinforcement, to formalise, and build upon, a student's existing knowledge models which have been constructed in more traditional educator-driven settings.

Indications from the usability questionnaires conducted during this evaluation were that the general usability of U-CREATe and the OCCS was positively rated. Both the students' opinion of the technical aspects of system usability and the students' attitude towards the educational experience were measured and produced generally positive results. These findings represent a positive indication of the usability of this research. However, the usability results still provide some room for improvement and it is hoped that further iterative design and evaluation can help to improve the overall performance of the system.

A feature of U-CREATe's design was the ability for a learner to tag each node in a mind map with links to specific pieces of content from the OCCS cache which they found to be informative or helpful in the exploration of that concept. These tags act as hyperlinks to the content in question which allows the learner to refer back to each piece of content when they revisit the mind map or revise the concepts in question. When the mind maps generated by the students during the trial were examined, less than 25% had included links to the content they found helpful in the OCCS when answering the questions. This may indicate that the students did not see the benefit of adding such links in relation to the educational experience. Only upon revisiting such maps or revising the concepts in question would the value of these links become apparent. However, these findings should not be dismissed and the benefit of this feature to the learner should be re-examined as part of the next design phase.

This evaluation, including both the pre and post-tests, was conducted in isolation, therefore any measured knowledge gain can be attributed to the process of completing the exercise in U-CREATe. However, long term knowledge gain is much more difficult to achieve, and also more difficult to attribute to one system or one method in isolation, as the students are interacting with other media and continuous learning is taking place in the meantime.

6.4 Summary of Third-Party Evaluation

In experiments conducted by another researcher, the OCCS has been successfully integrated in a dynamic hypertext generation framework [Steichen et al. 09]. This framework has been evaluated in an authentic learning environment with a class of undergraduate Computer Science students. This third-party evaluation produced further indicative results which reinforce the observations made in the experiment evaluations described above.

As detailed in section 2.3.4, Adaptive Hypermedia (AH) systems and, more specifically, Adaptive Educational Hypermedia Systems (AEHS) have traditionally focused on delivering personalised learning offerings to individual learners. The educational content used in the composition of such personalised offerings is typically sourced from a proprietary set of closed corpus resources. This inherit reliance upon handcrafted learning objects, enriched with considerable amounts of descriptive metadata, restricts the scalability of such systems. In an attempt to address such restrictions, the content sourcing capability of the OCCS was combined with a community-based content annotation interface and a dynamic hypertext generation system. This approach also aims to improve the variety of information available to the learner and to reduce the time and effort required by the educator in advance of learning offering generation.

The framework was evaluated with regard to educational benefit and learner satisfaction. Learner knowledge gain was the metric used to measure the educational effectiveness of the learning offering composed and delivered by the system. This was calculated in a similar fashion to the evaluation of U-CREATe, described above, with the use of a pre-test. The students' displayed a promising rate of knowledge gain and increased task completion, which indicates the benefits of the framework. An analysis of the usability questionnaire feedback illustrated the students generally positive attitude to the system and to the delivery of results in the form of adapted hypertext presentations.

While being very domain-specific in its implementation, this third-party framework and its evaluation provide an additional assessment of the performance of the OCCS content sourcing and cache-generation functionality. The results of this evaluation complement the key observations and considerations drawn from the OCCS evaluation described in the above sections.

6.5 Summary

This evaluation chapter endeavoured to address the goals and objectives of this thesis, as formed by the driving research question set forth in section 1.2 of this thesis. Two separate, yet complimentary trials were conducted which evaluated if the applications developed, and approaches adopted by this research satisfied these goals and objectives.

The first trial assessed the means of supporting the discovery, classification and harvesting of educational content from open corpus sources implemented in the OCCS and critiqued the relevance, quality and accuracy of the content harvested. The second trial examined the methods implemented in U-CREATe to support educators and learners in the exploration, incorporation and resource-level reuse of open corpus content in a pedagogically meaningful TEL experience.

The participants involved in both evaluation trials were selected as being exemplary of the profile of user who would utilise these applications when available for everyday use. The OCCS tool-chain was designed to facilitate course developers and educators in the discovery of relevant content which could subsequently be used in their TEL offerings. As such, the trial participants selected to take part in the evaluation were educators, familiar with TEL, who were also subject matter experts in the area of SQL. U-CREATe was designed as a demonstrator educational application which could deliver a pedagogically beneficial, learner-driven educational offering using content provided by the OCCS. Therefore, undergraduate students beginning to explore the area of SQL in their courses were selected as the trial participants for the second trial. This allowed accurate deductions to be made regarding the performance of U-CREATe based upon students' performance in the exercise and perception of the application.

The results presented in this chapter provide confidence in the ability of the OCCS to discover, classify, harvest and deliver highly relevant open corpus content in defined subject areas. The results also provide confidence that the resource-level reuse of such content in educational offerings is possible, and that these educational offerings can be pedagogically beneficial to the learner.

7 Conclusions

7.1 Introduction

This thesis presented research conducted into *leveraging content from open corpus sources for technology enhanced learning*. This innovative approach proposed the reuse of content from open corpus sources in educational offerings provided by TEL environments. The facilitation of which was realised through the development of a tool-chain that enables the discovery, classification, harvesting and delivery of content from the WWW, and a novel TEL application which demonstrates the resource-level reuse of open corpus content in the execution of a pedagogically meaningful educational offering.

The purpose of this chapter is to discuss the goals and objectives of this thesis and to illustrate how they have been achieved. It also identifies the contributions of this research to the state of the art of technology enhanced learning. Finally, the chapter concludes with a discussion of the possible future work which may be conducted to extend the research detailed in this thesis.

7.2 Objectives and Achievements

The research question posed in chapter one this thesis was to identify *the appropriate techniques and technologies required (i) to support the discovery, classification and harvesting of educational content from open corpus sources and (ii) to support educators and learners in the exploration, incorporation and reuse of such content in pedagogically meaningful TEL experiences*. Based on this question, five core research objectives were defined within two distinct research areas, namely:

1. *The Discovery, Harvesting, Classification and Delivery of Open Corpus Content*
    a. *To investigate and specify techniques and technologies which can be used to enable the dynamic discovery, classification and harvesting of content residing on the WWW and in certain defined digital content repositories.*
    b. *To design and develop an application, based on these techniques and technologies, which enables the creation of caches of educational content, sourced from the WWW, defined by subject area.*

c. *To investigate and implement the technologies required to index such open corpus content so that it is searchable and accessible to learners and educators.*

2. *TEL Application to Demonstrate the Resource-Level Reuse of Open Corpus Content*
   a. *To identify the learning theories and technologies required to support pedagogically beneficial TEL experiences in an interactive, learner-driven TEL application.*
   b. *To design and develop a demonstrator educational application which enables the learner-driven exploration and resource level reuse of content provided by the open corpus content service.*

In addressing objective 1.a, *"to investigate and specify techniques and technologies which can be used to enable the dynamic discovery, classification and harvesting of content residing on the WWW and in certain defined digital content repositories"*, an integral part of this research involved conducting a state of the art review of the Information Retrieval (IR) techniques and technologies in use on the WWW.

This review identified the various means of content discovery in use on the WWW with a specific focus on web crawling. Topic-specific, or focused web crawling was examined in particular detail, as a means of sourcing collections of content from the WWW in defined subject domains. This was an essential feature of the OCCS and it was decided to implement a focused crawling component as part of the tool-chain. A study of the link structure of the WWW was conducted, with particular focus on webometrics, which is the application of bibliometric methods to the study of the web. This was essential when designing a seeding mechanism for the OCCS. The links provided to a crawler dictate what content can be reached and are of paramount importance as illustrated by theories such as Hubs and Authorities [Kleinberg 99b], BowTie Graph Theory [Broder et al. 00] and Topical Locality [Davison 00]. Once suitable content has been identified, it must be represented in a manner which allows the automatic comparison and selection of content in response to a search query. The various facets of content indexing were reviewed, including document pre-processing, term-frequency analysis and term weighting. A number of IR models used to conduct query-document similarity calculation were then examined and the benefits of each approach detailed.

As it is goal of this research to utilise open source software where possible, a number of open source solutions to each of the necessary IR techniques were examined and discussed. This resulted in the selection of a Heritrix, Rainbow, JTCL, NutchWAX and WERA as components of the OCCS tool-chain. This state of the art review, which achieves objective 1.a, is documented in both Chapter 3 and Appendix B of this thesis and had a direct influence on the achievement of objectives (1.b) and (1.c).

In addressing objectives 1.b, *"to design and develop an application, based on these techniques and technologies, which enables the creation of caches of educational content, sourced from the WWW, defined by subject area"*, and 1.c, *"to investigate and implement the technologies required to index such open corpus content so that it is searchable and accessible to learners and educators"*, a design and specification was formulated for a content retrieval tool-chain which could traverse the web in search of content, identify content which falls within the scope of a particular subject area, harvest this content and make it discoverable via a search interface. This specification was disaggregated to provide the technical requirements for the Open Corpus Content Service (OCCS). The OCCS provides a suite of tools which can enable a non-technical educator to define the scope of a web crawl and generate caches of subject-specific open corpus content which can be exploited in TEL scenarios. The design and implementation of the OCCS is documented in sections 4.2, 4.3 and 5.2 and the tool-chain is evaluated in section 6.3.

The evaluation of the OCCS cache-generation conducted in chapter 6, provides confidence that the OCCS tool-chain generates collections of relevant, high-quality resources for an educator-defined subject domain. A manual examination of portions of the cache to identify the relevance of the resources and the IR metrics used to measure the performance of the IR application provide evidence that objective 1.b has been achieved. The performance of the indexing and searching functionality of the OCCS is also evaluated by these IR metrics. While the baseline requirements of objective 1.c have been satisfied by the learner being able to search for, and discover relevant material, the performance of this component of the OCCS could be improved. This would help to limit the dilution of result sets with irrelevant material. This should be examined in future development and refinement of the OCCS.

The OCCS was designed to adopt a caching approach to the harvesting and delivery of content. Content reuse is one of the key motivations for this research and this strategy aimed to make the OCCS itself, as flexible and reusable a service as possible. The caches of content generated by the tool-chain can be used in a wide variety of scenarios. Such caches can be integrated directly in a TEL offering, as they are in U-CREATe. The student can search for, and browse, informative content in their area of study whilst the educator has confidence that only content within the scope they have defined is accessible to the learner. This would provide many of the benefits the WWW has to offer as a learning resource, while minimising the risk of learner distraction or cognitive overload which can be encountered on the open WWW. It is also proposed that the OCCS could act as a feeder service for content analysis and repurposing services within TEL. Such applications could be used to adapt the content harvested from the web for integration into other TEL environments, such as Adaptive Educational Hypermedia Systems (AEHS).

However, the caching approach has not been without drawbacks, as discussed in section 6.2.6. As the hyperlinks on pages browsed in the OCCS do not link back to the open WWW, it can prove frustrating for learners who encounter links to relevant content which cannot be reached. This problem is particularly acute in the case of index pages. Issues have also arisen with the display of pages that contain embedded or interactive content in the OCCS. The usefulness of such pages is greatly reduced when such features no longer function as intended. A potential compromise approach to addressing this problem is to use the index of the cache when performing a search in the OCCS, then link to the live content on the WWW using the bounded, subject-specific cache as an entry point. This would involve a trade-off in terms of the elimination of user-distraction and the potential for a learner to browse to irrelevant content. It would also require the careful implementation of a URI revisiting policy in the crawler, to ensure that the cache index is up to date. An examination of the caching approach used and the link-handling implemented in the OCCS will be conducted during the next OCCS development cycle.

In addressing objective 2.a, *"to identify the learning theories and technologies required to support pedagogically beneficial TEL experiences in an interactive, learner-driven TEL application"*, an analysis of educational theories and how they are applied to ensure pedagogical effectiveness in technology enhanced learning was completed. A theoretical

categorisation of learning was examined which hypothesises that pedagogical approaches can be divided into three broad, overlapping perspectives: the associationist or empiricist perspective; the cognitive perspective; and the situative perspective. The design of TEL applications is often influenced by these pedagogical approaches and can display traits of more than one perspective. Section 2.2 also identified means of mapping educational theory to the pedagogical design of learning environments.

This analysis of educational theories and their application in TEL directly influenced the means by which objective 2.b was achieved. It reinforced that fact that the TEL application developed by this research should primarily address educational concerns and deliver not only engaging, but pedagogically meaningful learning experiences. It also identified that the evaluation of the TEL application should primarily focus on educational outcomes rather than technological performance. This analysis, which achieves objective 2.a, is documented in both Chapter 2 and Appendix A.

In addressing objective 2.b, *"to design and develop a demonstrator educational application which enables the learner-driven exploration and resource level reuse of content provided by the open corpus content service"*, an essential part of this research was to conduct an analysis of the current methods of educational content creation, dissemination and utilisation, with a particular focus on TEL. This analysis examined the current, and rapidly evolving, means by which content is authored and disseminated, with particular focus on the WWW and content in the education domain. The publication of content is becoming more accessible to the average individual and the volume of content available is increasing exponentially as a result. It is becoming more difficult to filter through content and identify relevant, high-quality resources. Standards and approaches which aim to formalise and describe educational content with the intention of increasing content portability and reuse were examined. These include learning objects, LOM and SCORM. The benefits, and limitations, of these approaches were examined. The emergence of digital content repositories in the education domain as a means of sharing and collating content was analysed. A mixture of cultural, legal and organisational issues have resulted in these initiatives facing problems with engagement. In addition, this analysis described the current methods of educational content utilisation in TEL applications including personalised or adaptive TEL systems. This analysis, which helped to achieve objective 2.b, is documented in section 2.3.

A design and specification was formulated for a TEL application which could utilise open corpus content in a learner-driven, pedagogically beneficial educational offering. This specification was disaggregated to form the educational, technical and usability requirements for the User-driven Content Retrieval, Exploration and Assembly Toolkit for eLearning (U-CREATe). U-CREATe provides a mind mapping interface which allows the learner to formalise and reflect their existing knowledge of a subject area and expand this knowledge using open corpus content provided by the OCCS. The design and implementation of U-CREATe has been documented in sections 4.2, 4.4 and 5.3 and the application is evaluated in section 6.3.

U-CREATe demonstrates a validated approach to the resource-level reuse of open corpus content in a pedagogically beneficial learning offering. The learner-driven offerings delivered by U-CREATe have been shown to contribute to knowledge gain in the majority of learners. However, as previously stated in section 4.4.1, the educational offerings made available by U-CREATe are not the only manner in which open corpus content can be reused. On the contrary, it is hoped that the OCCS can act as a feeder service for many diverse TEL applications. The pedagogical design of such applications and learning offerings is the responsibility of the TEL designer; U-CREATe merely demonstrates that such reuse is pedagogically valid, realistic and achievable.

7.3 Contribution to the State of the Art

A novel approach to the sourcing of open corpus content for integration and reuse in TEL is the primary contribution to the State of the Art made by this thesis and the research described therein. This approach is significantly different to that used by current TEL systems in the creation of learning offerings and provides a service which differs considerably from that offered by general purpose web search engines.

The OCCS provides a means of discovering and collating related content resources from open corpus sources. While web-based content is incorporated in TEL offerings in individual educational scenarios, the scale of open corpus content available remains largely unexploited in TEL. It is not common practice for educators to generate large collections of related content which can be utilised by TEL applications in the scalable delivery of learning

offerings in mainstream education. The OCCS provides the non-technical educator with the ability to define the scope of a subject area using simple inputs and receive a cache of subject specific content which can be used in a variety of scenarios. Content residing on the WWW is made available to TEL applications through a novel tool chain which seamlessly integrates methods of content discovery, language identification, subject classification, harvesting and storage. Subject-specific content caches are indexed and made searchable via a web interface to enable exploration and utilisation by the learner or educator.

This approach differs from the current methods of content utilisation in TEL, as the majority of content used by these systems is still manually authored in advance of learning offering generation and in a proprietary format. The authoring of content for TEL has proven to be an expensive and time-consuming task. This approach is not scalable if TEL is to be accepted in mainstream education. As such, means must be developed of leveraging the vast amounts of existing resources and knowledge available via the WWW. The OCCS benefits the State of the Art by providing a means of sourcing this content and making it available for reuse in TEL offerings.

The service offered by the OCCS tool chain differs from that provided by web search engines, e.g. Google, Bing etc., in a number of ways. While general purpose search engines perform admirably in serving run time searches for information across the entire web, the broad coverage necessary inevitably leads to the dilution of search results with irrelevant material for all but the most expertly constructed queries. The OCCS can help to minimise this dilution of result sets through the creation of subject specific caches of content. Novice users can then search for information in these caches of content with a reduced risk of being presented incorrect or irrelevant material. This approach can also help to limit the risk of distraction during a TEL exercise which browsing the open web may cause.

Web search engines do not allow the apriori identification and collation of resources in specific subject areas. The ability of the OCCS to generate these collections of content enables further enhancement or repurposing of the resources to be conducted in advance of their use in TEL offerings. This could include the annotation of the resources with informative metadata or the semantic and structural analysis and slicing of content to reduce its granularity. Arguably the most beneficial enhancement offered by the OCCS over

traditional web search engines is the ability of the educator to define the scope of the content collection upon which searches can be performed. Through the classification training process, the educator can define what should be included in the content cache during the web crawl. This allows the educator to tailor a collection to a particular audience or create a collection of content to match a curriculum.

The second contribution of this research is in the area of content reuse. U-CREATe has demonstrated and validated that it is possible to dynamically reuse open corpus content in pedagogically beneficial TEL experiences. This application exploits the OCCS tool chain to enable the exploration and resource-level reuse of open corpus educational content in a learner-driven educational offering. U-CREATe was developed to reflect elements of various educational theories including Behaviourism, Cognitivism, Constructivism and Enquiry-based Learning. This approach differs from the current methods of content utilisation in TEL as the majority of the current generation of TEL environments are closed corpus in nature. Such environments rely upon repositories of bespoke, proprietary and sufficiently annotated educational resources. Those TEL systems which do enable the incorporation of content from external sources require this content to be manually sourced and thoroughly annotated before being integrated into the system. U-CREATe illustrates that it is possible for TEL environments to leverage open corpus educational content.

The influence of this research on the State of the Art is reflected by its direct contribution to nine publications in international journals, conferences and workshops. These publications can be found in Appendix G. The most significant of these publications are:

- Lawless, S., Hederman, L., Wade, V. "*Enhancing Access to Open Corpus Educational Content: Learning in the Wild*", In the Proceedings of the 21st ACM Conference on Hypertext and Hypermedia, Hypertext 2008, Pittsburgh, PA, U.S.A. June 19th-21st, 2008.
- Lawless, S., Hederman, L., Wade, V. "*OCCS: Enabling the Dynamic Discovery, Harvesting and Delivery of Educational Content from Open Corpus Sources*", In the Proceedings of the 8th IEEE International Conference on Advanced Learning Technologies, I-CALT 2008, Santander, Cantabria, Spain. July 1st-5th, 2008.

- Dagger, D., O'Connor, A., Lawless, S., Walsh, E., Wade, V. "*Service Oriented TEL Platforms: From Monolithic Systems to Flexible Services*", In IEEE Internet Computing, Special Issue on Distance Learning, V. Wade & H. Ashman (eds.), vol. 11(3), pp. 28-35. May-June, 2007.
- Lawless, S. & Wade, V. "*Dynamic Content Discovery, Harvesting and Delivery, from Open Corpus Sources, for Adaptive Systems*", In the Proceedings of the Fourth International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems, AH 2006, V. Wade, H. Ashman, B. Smyth (eds.), Dublin, Ireland, LNCS 4018, Springer-Verlag, pp. 445–451. June 20th-23rd, 2006.
- Lawless, S., Dagger, D., Wade, V. "*Towards Requirements for the Dynamic Sourcing of Open Corpus Learning Content*", In the Proceedings of the World Conference on E-Learning in Corporate, Government, Healthcare, and Higher Education, E-Learn 2006, Honolulu, Hawaii, USA, T.C. Reeves & S.F. Yamashita (eds.). October 13th-17th, 2006.
- Lawless, S., Wade, V., Conlan, O. "*Dynamic Contextual TEL - Dynamic Content Discovery, Capture and Learning Object Generation from Open Corpus Sources*", In the Proceedings of the World Conference on E-Learning in Corporate, Government, Healthcare and Higher Education, E-Learn 2005, Vancouver, B.C., G. Richards (ed.), AACE, pp. 2158 – 2165. October 24th-28th, 2005.

7.4 Future Work

There are a number of areas in which there is potential for the research described in this thesis to be extended and advanced. Some of these areas are identified and discussed below and the potential extensions to the tools developed by this research to address these issues are described.

7.4.1 OCCS Service-Oriented Approach

The tools in the OCCS chain developed by this research should be as accessible as possible to non-technical educators, to allow them generate caches of content which can be used in their own specific TEL scenarios. An improved, intuitive user interface to facilitate interactions with the classification training process of the OCCS would be beneficial. The creation of such an interface could be completed as part of a more general move to a service-oriented

architectural design for the tool chain. Currently, the individual tools in the OCCS chain are quite tightly coupled as a result of format dependencies between services.

Reuse is an important theme in this research, and this reuse could be extended to the OCCS tools as well as the content upon which they operate. It is desirable that the components of the OCCS be accessible for use, not only through human interactions, but also by external applications. The individual OCCS components could be decoupled and exposed as web services through the use of the Web Services Description Language (WSDL) [WSDL].

WSDL is an XML format which is used to describe network services as a set of endpoints. An endpoint is defined by associating a network address with a reusable binding. A collection of such endpoints defines a service. A message is a description of the data to be exchanged and an endpoint type is a collection of operations which can be performed upon that data. A reusable binding is an endpoint type which has an associated concrete network protocol and data format specification defined. Operations and messages in the service are then bound to this network protocol and message format. WSDL uses these definitions to describe the public interface to the web service.

In this way, any external application which wishes to connect to an OCCS service can read the WSDL descriptions to determine what functions are available on the server. The client can then use Simple Object Access Protocol (SOAP) [SOAP] messages to call one of the available functions listed in the WSDL. An API such as JAX-WS [JAX-WS] or Axis2 [Axis2] could be used to construct the SOAP messages. These APIs convert service calls and matching replies to and from SOAP. This hides some of the complexity of communication between the web service and the client. An example web service architecture for the OCCS is displayed in figure 7-1, below. This architectural approach also provides the potential for the development of future services which can be easily plugged into the OCCS tool-chain.

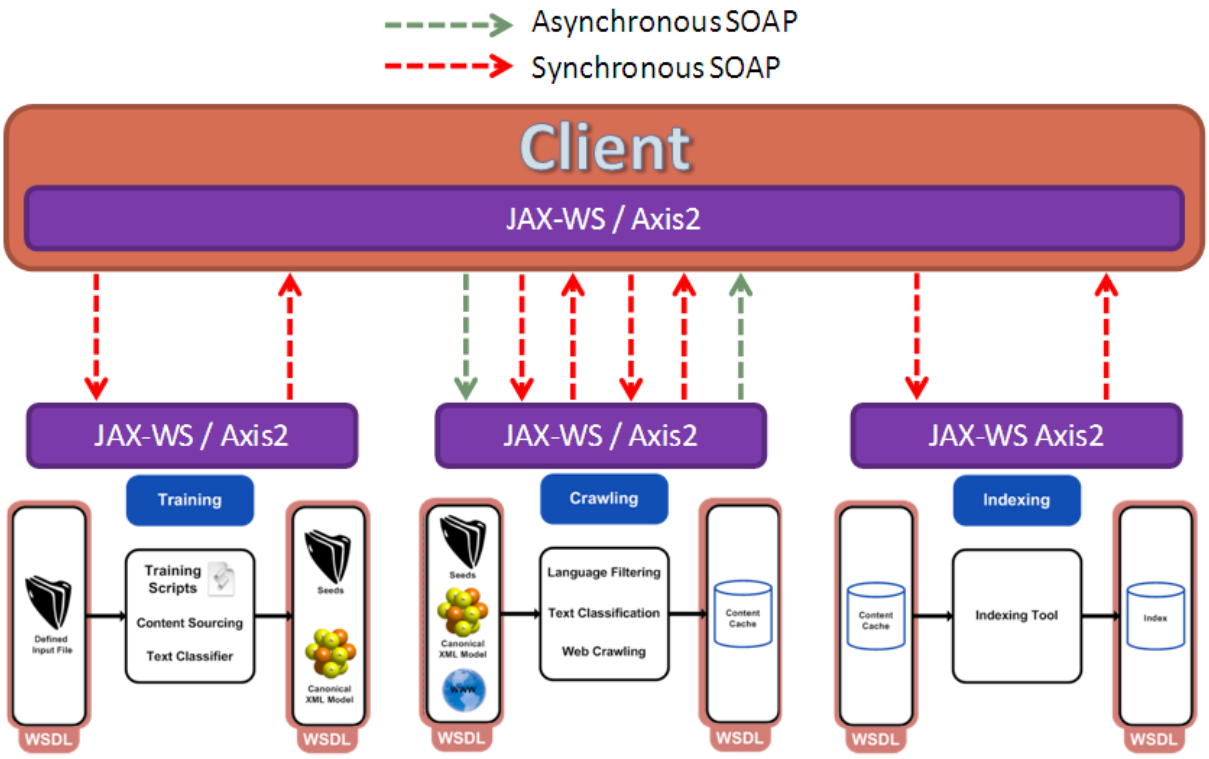


Figure 7-1 Proposed OCCS Service-Oriented Architecture

7.4.2 Incremental Crawling

Incremental or continuous crawls are web crawls which revisit previously fetched URIs to update a content cache or index to include changes made to content since the initial crawl. Implementing a page revisiting strategy would ensure that the content provided to the educator for use in a learning offering remained current and correct throughout its period of use. Even if a caching approach was not used, page revisiting would ensure that the index representation of a page was as accurate as possible to aid resource discovery.

There are numerous means of implementing crawl revisiting strategies. Heritrix has a continuous crawling strategy which is described in section 5.2.3. This allows the crawler to "revisit previously fetched pages, looking for changes, as well as discovering and fetching new pages, even adapting its rate of visitation based on operator parameters and estimated change frequencies."

The sitemaps protocol allows web server administrators to inform a web crawler of new or updated content on their site [Schonfeld & Shivakumar 09]. In its most basic form, a sitemaps

file is an XML file that contains a list of URLs for a website. The file can also contain information relating to these URLs, such as last modification time, expected change frequency and priority. This information allows the web crawler to devise the most efficient URL revisiting strategy for each site. While not yet fully adopted, sitemaps are growing in popularity with over 35 million websites currently publishing sitemap files [Schonfeld & Shivakumar 09]. Implementing sitemap support in the OCCS would help to improve content discovery and efficiently maintain content caches and indexes.

7.4.3 Automated Assessment of Content Collections

As discussed in section 6.2, it is infeasible to manually examine an entire cache of content which could potentially contain thousands, or even millions, of resources. As a result it was decided to manually examine a portion of the cache to ensure that the content was relevant, technically valid and of sufficient quality to be included in an educational offering. However, this manual review process could be supplemented by an automatic analysis of the entire content cache in an attempt to identify off-topic or irrelevant resources in advance.

There are numerous statistical approaches which could be used to perform such automated assessments of relevance. Corpus analysis can be used to identify unusual or outlying resources in a collection. Each resource is represented as a feature vector based upon its content. Cosine similarity can then be used to deduce n-dimensional distances between the resources and identify outliers which can be removed from the collection. Clustering algorithms can be used in a similar fashion to identify unfit resources in a collection. The cache is modelled as a graph of nodes, with the edges between these nodes weighted using language modelling techniques. Graph analysis can then be used to identify and remove unfit documents from the collection.

7.4.4 Additional Evaluation

As discussed in sections 6.2.6 and 7.2, the OCCS implemented a caching approach to the discovery and collation of subject-specific web content. This approach was adopted with the aim of making the OCCS as flexible and reusable a service as possible. The caches generated could be used directly or as inputs to other content analysis and repurposing services within TEL. Through the provision of controlled, subject-specific collections of content, the risk of content dilution and learner distraction is minimised.

However, this caching approach has some disadvantages. Inactive or broken hyperlinks to potentially relevant content can prove frustrating to the learner. This problem is particularly acute in index pages which contain many valuable links. Rendering problems with embedded or interactive content have also become apparent. An examination of the caching approach and link-handling which has been implemented in the OCCS will be conducted during the next OCCS development cycle.

Currently the OCCS provides a different service than that offered by conventional search engines which do not provide tailored means of discovering, classifying and harvesting content in particular subject areas. However, should the OCCS adopt a more conventional, search engine approach of redirecting the learner back to the open WWW then comparative evaluations with similar tools, such as Google, will be necessary.

A learner should be more likely to find relevant resources if they use the OCCS as the search is conducted across a set of predominantly relevant material. As a result, there should be less dilution of results. While it is not an aim of this research to compare the performance of the NutchWAX query-document similarity algorithm with Google's proprietary query-document similarity algorithm, it would be necessary to compare the service provided to the learner by each approach.

7.4.5 Content Slicing and Composition

As described in section 2.3.1, there is an inverse relationship between the potential reusability of a piece of educational content and its granularity. The larger and more complex a resource is, the more contextually specific it tends to be and the more difficult reuse becomes. Fine-grained, conceptually atomic, pieces of educational content are much more easily reused outside of the context for which they were created. Much research is being conducted into methods of content slicing and disaggregation. Structural and linguistic analysis can be combined to develop intelligent content slicing technologies. Structural segmentation techniques include densitometric analysis [Kohlschütter & Nejdl 09], DOM tree pattern analysis [Vieira et al. 06], isotonic regression [Chakrabarti et al. 07], vision-based techniques [Baluja 06] [Cai et al. 06] and token-based approaches [Pasternack & Roth 09]. Lexical segmentation approaches include the use of supervised learning techniques such as hidden markov models, dictionary and rule-based approaches and word-sense clustering.

A service could be added to the OCCS tool-chain which performs such slicing upon the content contained in each subject-specific cache. This would result in a collection of subject-specific, conceptually-atomic content objects which are more fine-grained and are potentially more easily reusable by TEL environments in the generation of educational offerings. The structural and lexical analysis conducted by such a service could also be utilised to perform a level of semantic annotation of content.

Content slicing and annotation functionality would provide the opportunity for the development of novel approaches in content composition. Open corpus content which has been harvested, sliced and annotated, must still be returned to the user in a manner which satisfies their information need. The dynamic composition of fine-grained content objects could be used to provide personalised intelligent responses.

7.4.6 DRM and IPR

While deemed outside the scope of this thesis, issues surrounding Digital Rights Management (DRM) and Intellectual Property Rights (IPR) will have to be addressed before the OCCS and TEL applications like U-CREATe are deployed in mainstream education. As discussed in section 2.3.5, uncertainties and misconceptions in relation to copyright law, ownership and intellectual property pose arguably the most significant barrier to the mainstream adoption of educational content sharing initiatives.

Initiatives such as Creative Commons [Creative Commons] have begun to make significant strides in this area through the embedding of licenses within content. Through the simple addition of Creative Commons metadata to content, an author can assert their rights over that content. Four license types are supported by Creative Commons: Attribution, Share Alike, Noncommercial and No Derivative Works,

A crawl configuration option could be provided by the OCCS in a similar fashion to its politeness policy. Before crawling a site the robots.txt file is examined and the authors wishes with relation to downloading are respected. A similar policy could be implemented with respect to Creative Commons. Content harvesting could be made dependent upon its associated license type. The educator could specify in advance of the crawl which license types they are willing to accept. The educator may only want content which they can modify for their own purpose, so they would specify Attribution, Share Alike and Noncommercial

licence types. This process would go some way to addressing the concerns of content authors and reusers alike, who wish to use the OCCS tool-chain and TEL applications like U-CREATe.

Bibliography

[AceWiki] AceWiki is a semantic wiki which makes use of a conrtollef natural language called ACE. Available online at http://attempto.ifi.uzh.ch/acewiki/

[ADL] Advanced Distributed Learning. An American Government initiative involving both the public and private sectors. Its aim is to develop standards, tools and learning content for future learning environments. Available online at http://www.adlnet.gov

[Aggarwal et al. 01] Aggarwal, C.C., Al-Garawi, F., Yu, P.S. "Intelligent Crawling on the WWW with Arbitrary Predicates". In Proceedings of 10th International World Wide Web Conference, WWW2001, Hong Kong, China, May, 2001.

[Agosti & Melucci 00] Agosti, M. & Melucci, M. "Information Retrieval on the Web". In the European Summer School on Information Retrieval, ESSIR, Lecture Notes on Computer Science, vol. 1980, pp. 242-285, Springer-Verlag. 2000.

[Albright 05] Albright, P. "Open Educational Resources, Open Content for Higher Education: Final Forum Report". UNESCO International Institute for Educational Planning. October – December, 2005.

[Alexandria] The Library of Alexandria in Egypt, once the largest library in the ancient world, is generally regarded as the first attempt to gather all of humanities knowledge in a single location. More detail available online at http://en.wikipedia.org/wiki/Library_of_Alexandria

[Almind & Ingwersen 97] Almind, T. & Ingwersen, P. "Informetric analyses on the World Wide Web: Methodological approaches to 'webometrics'". In Journal of Documentation, vol. 53(4), pp. 404-426, 1997.

[AltaVista] AltaVista, a search engine from Overture Services, Inc. Available online at http://www.altavista.com/

[Anderson et al. 01] Anderson, L., Krathwohl, D., Airasian, P., Cruikshank, K., Mayer, R., Pintrich, P., Raths, J., Wittrock, M. "A Taxonomy for Learning, Teaching and Assessing: A Revision of Bloom's Taxonomy of Educational Objectives: Complete Edition". Longman: New York. 2001.

[Ardö 05] Ardö, A. "Focused Crawling in the Alvis Semantic Search Engine". In Proceedings, 2nd Annual European Semantic Web Conference (ESWC 2005), Heraklion, Crete, Greece, 29th May – 1st June 2005.

[Aroyo et al. 03] Aroyo, L., De Bra, P., Houben, G.J. "Embedding Information Retrieval in Adaptive Hypermedia: IR meets AHA!". In the Proceedings of the Workshop on Adaptive Hypermedia and Adaptive Web-Based Systems, at the Twelfth International World Wide Web Conference, WWW2003, pp. 63-76, Budapest, Hungary. May 20th, 2003.

[Ask] Ask, a search engine formally known as Ask Jeeves. Available online at http://www.ask.com/

[ASL] Software Licensing from the Apache Foundation. Terms available online at: http://www.apache.org/licenses/LICENSE-2.0

[Ausubel 00] Ausubel, D. P. "The Acquisition and Retention of Knowledge: A Cognitive View". Kluwer Academic Publishers, Norwell, MA, USA. 2000.

[Bailey et al. 00] Bailey, P., Craswell, N., Hawking, D. "Dark Matter on the Web". In Poster Proceedings of the 9th International Conference on the World Wide Web, WWW9, Amsterdam, Netherlands, 2000.

[Bailey et al. 06] Bailey, C., Zalfan, M.T., Davis, H.C., Fill, K. & Conole, G. "Panning for Gold: Designing Pedagogically-Inspired Learning Nuggets". In Educational Technology and Society, vol. 9(1), pp. 113-122. 2006.

[Baird et al 06] Baird, K. & The Jorum Team. "Automated Metadata: A Review of Existing and Potential Metadata Automation Within Jorum and an Overview of Other Automation Systems". JISC Project Report. March, 2006.

[Baluja 06] Baluja, S. "Browsing on Small Screens: Recasting Web Page Segmentation into an Efficient Machine Learning Framework". In the Proceedings of the 15th International World Wide Web Conference, WWW2006, pp. 33-42, Edinburgh, Scotland. 23rd-26th May, 2006.

[Bancroft, 01] Bancroft, D. in Healy, Y. "Caution on E-learning". In The Irish Times – Education and Living. Tuesday 17th April 2001. Article is available online at http://www.irishtimes.com/newspaper/education/2001/0417/01041700175.html - Last accessed 17th September 2008.

[Barab & Duffy 99] Barab, S.A. & Duffy, T.M. "From Practice Fields to Communities of Practice". In Theoretical Foundations of Learning Environments, D.H. Jonassen & S.M. Land (Eds.), Lawrence Erlbaum: NJ, U.S.A. 1999.

[Bates et al. 06] Bates, M., Loddington, S., Manuel, S., Oppenheim, C. “Rights and Rewards Project, Academic Survey: Final Report”. Final Report of JISC Funded Rights and Rewards Project. January, 2006.

[Bates et al. 07] Bates, M., Loddington, S., Manuel, S., Oppenheim, C. “Attitudes to the Rights and Rewards for Author Contributions to Repositories for Teaching and Learning”. In ALT-J, Research in Learning Technology, vol. 15(1), pp. 67-82. 2007

[Battelle 05] Battelle, J. “The Search: How Google and It’s Rivals Rewrote the Rules of Business and Transformed Our Culture”. Boston, MA; London: Nicholas Brearley Publishing, 2005.

[Bayes 1763] Bayes, T. “An Essay Towards Solving a Problem in the Doctrine of Chances”. In The Philosophical Transactions of the Royal Society: Giving Some Account of the Present Undertakings, Studies and Labours of the Ingenious in Many Considerable Parts of the World, vol. 53, pp. 370–418. 1763.

[Bing] A search engine from Microsoft. Available online at http://www.bing.com

[Beetham & Sharpe 07] Beetham, H. & Sharpe, R. “Rethinking Pedagogy for a Digital Age: Designing and Delivering E-Learning”. Routledge: New York, NY, USA. April, 2007.

[Bergman 01] Bergman, M.K. “The Deep Web: Surfacing Hidden Value”. In the Journal of Electronic Publishing, vol. 7(1), August, 2001.

[Berners-Lee et al. 92] Berners-Lee, T., Cailliau, R., Groff, J.F. and Pollermann, B. “World-Wide Web: The Information Universe”. Electronic Networking: Research, Applications and Policy, vol. 2(1), pp. 52-58. 1992.

[Berners-Lee et al. 01] Berners-Lee, T., Hendler, J., Lassila, O. “The Semantic Web”. Scientific American, pp. 35-43, May 2001.

[Bharat & Chang 03] Bharat, K. & Chang, B.W. "Web Search Engines: Algorithms and User Interfaces". Tutorial at the International Conference on Human Factors in Computing Systems, CHI 2003, Fort Lauderdale, FL, U.S.A. April 5th-10th, 2003.

[Biggs 07] Biggs, J. “Teaching for Quality Learning at University: What the Student Does”. 3rd Edition, Open University Press. 2007.

[Björneborn & Ingwersen 04] Björneborn, L., & Ingwersen, P. “Toward a Basic Framework for Webometrics”. In the Journal of the American Society for Information Science and Technology, vol. 55(14), pp. 1216-1227, John Wiley & Sons Inc., New York, USA. 2004.

[Bloom & Krathwohl 56] Bloom, B. & Krathwohl, D. “Taxonomy of Educational Objectives: The Classification of Educational Goals. Handbook 1: Cognitive Domain”. Longmans, Green: New York. 1956.

[Bordogna & Pasi 00] Bordogna, G. & Pasi, G. “Flexible Representation and Querying of Heterogeneous Structured Documents”. In Kibernetica, vol. 36(6), pp. 617-633. 2000.

[Boyle 03] Boyle, T. “Design Principles for Authoring Dynamic, Reusable Learning Objects”. In the Australian Journal of Educational Technology, vol. 19(1), pp. 46-58, 2003.

[BOW] A Toolkit for Statistical Language Modeling, Text Retrieval, Classification and Clustering. Available online at: http://www.cs.cmu.edu/

[Brady et al. 05] Brady, A., Conlan, O., Wade, V. “Towards the Dynamic Personalized Selection and Creation of Learning Objects”. In the Proceedings of the World Conference on E-Learning in Corporate, Government, Healthcare and Higher Education, E-Learn 2005, G. Richards (Ed.), pp. 1903–1909, Vancouver, B.C., Canada. November, 2005.

[Brin & Page 98] Brin, S. & Page, L. “The Anatomy of a Large-Scale Hypertextual Web Search Engine”. In Computer Networks and ISDN Systems, vol. 30(1-7), pp. 107-117, April 1998.

[Broder 02] Broder, A. “A Taxonomy of Web Search”. In SIGIR Forum, vol. 36(2), pp. 3-10. 2002.

[Broder et al. 00] Broder, A., Kumar, R., Maghoul, F., Raghavan, P., Rajagopalan, S., Stata, R., Tomkins, A., Wiener, J. “Graph Structure in the Web”. In Proceedings of the 9th International Conference on the World Wide Web, WWW9, Amsterdam, Netherlands, pp.309-320, 2000.

[Brooke 96] Brooke, J. “SUS: A "Quick and Dirty" Usability Scale”. In Usability Evaluation in Industry, P. W. Jordan, B. Thomas, B. A. Weerdmeester & A. L. McClelland (eds.). London: Taylor and Francis. 1996.

[Brown et al. 89] Brown, J.S., Collins, A., Duguid, P. “Situated Cognition and the Culture of Learning”. In Educational Researcher, vol. 18(1), pp. 32-41. 1989.

[Brown et al. 07] Brown, E., Fisher, T., Brailsford, T. “Real Users, Real Results: Examining the Limitations of Learning Styles within AEH”. Proceedings of the Eighteenth ACM Conference on Hypertext and Hypermedia, Hypertext 07, Manchester, UK, 10-12 Sept 2007.

[Brusilovsky 96] Brusilovsky, P. “Methods and Techniques of Adaptive Hypermedia”. In Special Issue on Adaptive Hypertext and Hypermedia, User Modeling and User-Adapted Interaction, P. Brusilovsky and J. Vassileva (eds.), vol. 6 (2-3), pp. 87-129. 1996.

[Brusilovsky 98] Brusilovsky, P. “Adaptive Educational Systems on the Worldwide Web: A Review of Available Technologies”. In the Proceedings of the Workshop "WWW-Based Tutoring", at the 4th International Conference on Intelligent Tutoring Systems, ITS'98, San Antonio, TX, USA. 1998.

[Brusilovsky 04a] Brusilovsky, P. “Adaptive Educational Hypermedia: From Generation to Generation”. In the Proceedings of the 4th Hellenic Conference with International Participation in “Information and Communication Technologies in Education”, M. Grigoriadou et al. (Eds), New Technologies Publications, pp. 19-33. 2004.

[Brusilovsky 04b] Brusilovsky, P. “KnowledgeTree: A Distributed Architecture for Adaptive E-Learning”. In the Proceedings of the Thirteenth International World Wide Web, WWW2004, Alternate track papers and posters, ACM Press, pp. 104–113, Manhattan, NY, USA. May 17th-20th, 2004.Conference,

[Brusilovsky & Henze 07] Brusilovsky, P. & Henze, N. “Open Corpus Adaptive Educational Hypermedia”. In The Adaptive Web: Methods and Strategies of Web Personalisation, Lecture Notes in Computer Science, vol. 4321, Berlin: Springer Verlag, pp. 671-696. 2007.

[Brusilovsky & Peylo 03] Brusilovsky, P. & Peylo, C. “Adaptive and intelligent Web-based educational systems”. International Journal of Artificial Intelligence in Education, 13(2-4), 159-172, 2003.

[Brusilovsky et al. 96] Brusilovsky, P., Schwarz, E., Weber, G. “ELM-ART: An intelligent tutoring system on World Wide Web”, In the Proceedings of the 3rd International Conference on Intelligent Tutoring Systems, ITS-96, C. Frasson, G. Gauthier, & A. Lesgold (Eds.), Lecture Notes in Computer Science, Vol. 1086, Berlin: Springer Verlag, pp. 261-269. 1996.

[Brusilovsky et al. 98] Brusilovsky, P., Eklund, J., Schwarz, E. "Web-based Education for All: A tool for Developing Adaptive Courseware". In Computer Networks and ISDN Systems, The Proceedings of the Seventh International World Wide Web Conference, vol. 30 (1-7), pp. 291-300. 14th-18th April, 1998.

[Brusilovsky et al. 04] Brusilovsky, P., Chavan, G., Farzan, R. "Social Adaptive Navigation Support for Open Corpus Electronic Textbooks". In Proceedings of 3rd International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems, AH 2004, P. DeBra, W. Nejdl (Eds.), Lecture Notes in Computer Science, Vol. 3137. Berlin: Springer Verlag, pp. 24-33. 2004.

[Bush 45] Bush, V. "As We May Think". In Atlantic Monthly (AM), July 1945. Available online at http://www.theatlantic.com/doc/194507/bush

[Buzan 74] Buzan, T. "Use Your Head". BBC: London.

[Buzan & Buzan 93] Buzan, T. & Buzan, B. "The Mind Map Book". Penguin Group: New York. 1993.

[Cai et al. 06] Cai, D., Shipeng, Y., Wen, J.R. & Ma, W.Y. "Extracting Content Structure for Web Pages based on Visual Representation". In the Proceedings of the 5th Asia Pacific Web Conference, APWeb 2003, pp. 406-417, Xi'an, China. 23rd-25th April, 2003.

[Carey 93] Carey, D. "Teacher Roles and Technology Integration: Moving from Teacher as Director to Teacher as Facilitator". Computers in the Schools, vol. 9(2-3), pp. 105-118. 1993.

[Capstick et al. 00] Capstick, J., Diagne, A.K., UszKoreit, H., Leisenberg, A., Leisenberg, M., Erbach, G. "A System for Supporting Cross-Lingual Information Retrieval". In the Journal of Information Processing and Management, vol. 36, pp. 275-289. 2000.

[Carlile et al. 04] Carlile, O., Jordan, A., Stack, A. "Learning by Design: Learning Theory for the Designer of Multimedia Educational Materials". WIT/BBC Online: Waterford, Ireland. 2004.

[Carlile & Jordan 05] Carlile, O. & Jordan, A. "It works in Practice but will it work in Theory? The Theoretical Underpinnings of Pedagogy". In Emerging Issues in the Practice of University Learning and Teaching, G. O'Neill, S. Moore, B. McMullin (Eds.), AISHE: Dublin, Ireland. 2005.

[Carmona et al. 02] Carmona, C., Bueno, D., Guzmán, E., Conejo, R. "SIGUE: Making Web Courses Adaptive". In Proceedings of 2nd International Conference on Adaptive Hypermedia and Adaptive Web Based Systems, AH2002, Malaga, Spain, 29-31 May, 2002. Lecture Notes on Computer Science, Vol. 2347. Berlin: Springer Verlag, pp. 376-379. 2002.

[Carter & Richardson 07] Carter, J. & Richardson, A. Transcription of an Interview with Jacqueline Carter and Andrew Richardson a summary of which appears in "Sharing TEL Content – A Synthesis and Commentary" by Ferguson et al. 07.

[Casey et al. 07] Casey, J., Proven, J., Dripps, D. "Managing Intellectual Property Rights in Digital Learning Materials: A Development Pack for Institutional Repositories". TrustDR Project Report. July, 2007.

[Cavnar & Trenkle 94] Canvar, W. B. and Trenkle, J. M. "N-gram based Text Categorization", Symposium on Document Analysis and Information Retrieval, University of Nevada, Las Vegas, p161-176, 1994.

[Chakrabarti et al. 99] Chakrabarti, S., van den Berg, M., Dom, B. "Focused Crawling: A New Approach to Topic-Specific Web Resource Discovery". In The International Journal of Computer and Telecommunications Networking, Vol. 31(11-16), Elsevier North-Holland, Inc. New York, NY, USA. pp. 1623-1640, May 1999.

[Chakrabarti et al. 02] Chakrabarti, S., Punera, K., Subramanyam, M. "Accelerated Focused Crawling through Online Relevance Feedback". In proceedings of the Eleventh International World Wide Web Conference, WWW2002, Honolulu, Hawaii, USA. May 7-11, 2002.

[Chakrabarti et al. 07] Chakrabarti, D., Kumar, R. & Punera, K. "Page-level Template Detection via Isotonic Smoothing". In the Proceedings of the 16th International World Wide Web Conference, pp. 61-70, Banff, Alberta, Canada. May 8th-12th, 2007.

[Charlesworth et al. 07] Charlesworth, A., Ferguson, N., Schmoller, S., Smith, N. & Tice, R. "Sharing TEL Content – A Synthesis and Commentary". London: JISC. Available online at http://ie-repository.jisc.ac.uk/46/1/selc-final-report-3.2.pdf. 2007.

[Charniak 97] Charniak, E. "Statistical Techniques for Natural Language Parsing". In AI Magazine, vol. 18, pp. 33-44. 1997.

[Chen et al. 98] Chen, H., Chung, Y., Ramsey, M., Yang, C. "A Smart Itsy Bitsy Spider for the Web". In the Journal of the American Society for Information Science, vol. 49(7), pp. 604-618, 1998.

[Cho et al. 98] Cho, J., Garcia-Molina, H., Page, L. "Efficient Crawling Through URL Ordering". In Proceedings of the Seventh World Wide Web Conference, WWW7, Brisbane, Australia, April 14-18, 1998. Also in Computer Networks and ISDN Systems, vol. 30(1-7), pp. 161-172. 1998.

[CLEVER] The CLEVER Search Engine. An IBM Research Project. Detail available online at http://www.almaden.ibm.com/projects/clever.shtml

[CMU OLI] Carnegie Mellon University's Open Learning Initiative. Available online at http://www.cmu.edu/oli/

[Combine] Combine Web Crawler is an open source system for general and focused web crawling and indexing. Available online at http://combine.it.lth.se/

[comScore] comScore Search Engine Market Share Report – June 2008. Available online at http://www.comscore.com/press/data/share_of_search.asp

[Collins & Moonen 01] Collis, B. & Moonen, J. "Flexible learning in a digital world: Experiences and Expectations". London: Kogan Page. 2001.

[Conklin, 87] Conklin, J. "Hypertext: An Introduction and Survey". IEEE Computer, 20(9), pp. 17-41. 1987.

[Conlan 05] Conlan, O. "The Multi-Model, Metadata Driven Approach to Personalised TEL Services". Doctoral Thesis, Submitted to the University of Dublin, Trinity College, 2005.

[Conlan et al., 02a] Conlan, O., Wade, V., Bruen, C., Gargan, M. "Multi-Model, Metadata Driven Approach to Adaptive Hypermedia Services for Personalized TEL". In the Proceedings of the Second International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems, AH2002, Malaga, Spain. May, 2002.

[Conlan et al. 02b] Conlan, O., Hockemeyer, C., Wade, V., Albert, D. "Metadata Driven Approaches to Facilitate Adaptivity in Personalized TEL Systems". In the Journal of the Japanese Society for Information and Systems in Education, vol. 1(1), pp. 38–45. 2002.

[Conlan & Wade 04] Conlan, O. & Wade, V. “Evaluation of APeLS - An Adaptive TEL Service based on the Multi-model, Metadata-driven Approach”. In the Proceedings of the Third International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems, AH2004, P. De Bra & W. Nejdl (Eds.), Berlin: Springer Verlag, pp. 291–295. 2004.

[Connexions] An open repository project for the sharing and reuse of educational material. Available online at http://www.cnx.org

[Coombs 1990] Coombs, J.H. “Hypertext, Full Text, and Automatic Linking”. In Proceedings of the 13th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Brussels, Belgium. pp 83-98, September 5th-7th, 1990.

[CPAN] The Comprehensive Perl Archive Network, a large collection of Perl software and documentation. Available online at: http://www.cpan.org/

[Cranky] Cranky.com the first age-relevant search engine. Available online at: http://www.cranky.com

[Creative Commons] Creative Commons is a licensing standard which provides a flexible range of protections and freedoms for authors, artists, and educators. Available online at http://creativecommons.org/

[Cristea & Carro 07] Cristea, A., Carro, R. “Authoring of Adaptive and Adaptable Hypermedia: An Introduction”. In the International Journal of Learning Technology, Special Issue on Authoring of Adaptive and Adaptable Hypermedia, vol. 3(3), Inderscience. 2007.

[Cronin 84] Cronin, B. “The Citation Process: The Role and Significance of Citations in Scientific Communication”. Taylor Graham, London. 1984

[Cummins & O’Riordan 06] Cummins, R. & O’Riordan, C. “Evolving Local and Global Weighting Schemes in Information Retrieval”. In Information Retrieval, vol. 9(3), pp. 311-330. June, 2006.

[Cuil] Cuil, claims to be the world’s largest search engine. Available online at: http://www.cuil.com

[Currier et al. 04] Currier, S., Barton, J., O’Beirne, R. & Ryan, B. “Quality Assurance for Digital Learning Object Repositories: Issues for the Metadata Creation Process”. In ALT-J: Research in Technology Learning, vol. 12(1), pp. 5-20. 2004.

[Cutler et al. 99] Cutler, M., Deng, H., Maniccam, S., Meng, W. "A New Study on using HTML Structures to Improve Retrieval". In Proceedings of the 11th IEEE International Conference on Tools with Artificial Intelligence, ICTAI 99, pp. 406-409, Chicago, Illinois, U.S.A. 8TH-10TH November, 1999.

[Dagger 06] Dagger, D. "Personalised TEL Development Environments", Doctoral Thesis, Submitted to the University of Dublin, Trinity College, 2006.

[Dagger et al. 04] Dagger, D., Wade, V., Conlan, O. "Developing Adaptive Pedagogy with the Adaptive Course Construction Toolkit (ACCT)". In the Proceedings of the Third International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems, AH2004, P. De Bra & W. Nejdl (Eds.), Berlin: Springer Verlag, pp. 55–64, Eindhoven, The Netherlands. August, 2004.

[Dagger et al. 05] Dagger, D., Wade, V., Conlan, O. "Personalisation for All: Making Adaptive Course Composition Easy". In Special Edition of the Educational Technology and Society Journal, IEEE IFETS, vol. 8(3), pp. 9-25. 2005.

[Dagger at al. 07] Dagger, D., O'Connor, A., Lawless, S., Walsh, E., Wade, V. "Service Oriented TEL Platforms: From Monolithic Systems to Flexible Services". In IEEE Internet Computing Special Issue on Distance Learning, vol. 11(3), pp 28-35, June 2007.

[Dai et al. 05] Dai, S., Diao, Q., Zhou, C. "Performance Comparison of Language Models for Information Retrieval". In Artificial Intelligence Applications and Innovations, vol. 187, pp.721-730, Springer Verlag: Boston, USA. 2005.

[Date 99] Date, C.J. "An Introduction to Database Systems", 7th Ed., Addison-Wesley Longman. Boston, MA. 1999.

[Davis et al. 93] Davis, H.C., Hutchings, G.A. & Hall, W. "Microcosm: A Hypermedia Platform for the Delivery of Learning Materials". Technical Report. 1993

[Davis et al. 07] Davis, H.C., Dibiase, D., Fill, K., Martin, D., Rees, M. "DialogPLUS: Final Report". Final Report of the JISC and NSF Funded DialogPLUS Project. January, 2007.

[Davis et al. 09] Davis, H.C., Carr, L., Hey, J.M.N., Howard, Y., Millard, D., Morris, D. & White, S. "Bootstrapping a Culture of Sharing to Facilitate Open Educational Resources". In IEEE Transactions on Learning Technologies. In Press. May, 2009.

[Davison 00] Davison, B.D. “Topical Locality on the Web”. In Proceedings of the 23rd International Conference on Research in Information Retrieval, SIGIR 2000, pp. 272-279, Athens, Greece. July, 2000.

[DCMI] The Dublin Core Metadata Initiative is an open organization engaged in the development of interoperable online metadata standards that support a broad range of purposes and business models. Available online at: http://dublincore.org/

[De Bra & Calvi 98] De Bra, P. & Calvi, L. “AHA: a Generic Adaptive Hypermedia System”. In the Proceedings of the 2nd Workshop on Adaptive Hypertext and Hypermedia, pp. 5-12, Pittsburgh. 1998.

[De Bra & Post 94b] DeBra, P. & Post, R. “Information Retrieval in the Worldwide Web: Making Client-based Searching Feasible”. In Computer Networks and ISDN Systems, vol. 27(2), pp. 183-192. 1994.

[De Bra & Post 94a] De Bra, P., Post, R. “Searching for Arbitrary Information in the WWW: The Fish-Search for Mosaic”. In Proceedings of 2nd International World Wide Web Conference, WWW94, P. Enslow, I. Goldstein, J. Hardin (Eds.), Chicago, U.S.A. Elsevier Science Publishers: Amsterdam. October 1994.

[De Bra et al. 94] De Bra, P., Houben, G., Kornatzky, Y., Post. R. “Information Retrieval in Distributed Hypertexts”. In Proceedings of the 4th Intelligent Multimedia Information Retrieval Systems and Management Conference, RIAO 94, Rockefeller University, New York, U.S.A. pp. 481-491. 1994.

[De Bra et al. 99] De Bra, P., Houben, G.J., Wu, H. “AHAM: A Dexter-based Reference Model for Adaptive Hypermedia”. In the Proceedings of the Tenth ACM Conference on Hypertext and Hypermedia, Hypertext 99, pp. 147-156, Darmstadt, Germany. 1999.

[De Bra et al. 03] De Bra, P., Aerts, A., Berden, B., De Lange, B., Rousseau, B., Santic, T., Smits, D., Stash, N. “AHA! The Adaptive Hypermedia Architecture”. In the Proceedings of the Fourteenth ACM Conference on Hypertext and Hypermedia, pp. 81–84, Nottingham, England. August 26th-30th, 2003.the

[De Carvalho Fontes & Silva 04] De Carvalho Fontes, A., Silva, F.S. “Smartcrawl: A New Strategy for the Exploration of the Hidden Web”. In Proceedings of the 6th ACM International Workshop on Web Information and Data Management, WIDM 04, pp. 9–15, New York, NY, USA. 2004.

[Deepak & Parameswaran 05] Deepak P. & Parameswaran, S. “Features in the Web Search Interface: How Effective are They?”. In Proceedings of the International Conference on Multilingual Computing and Information Management in Networked Digital Environments, CALIBER 2005, Kochi, India. Feb 2-4, 2005

[DFES 03] Department for Education and Skills. “Towards a Unified e-learning Strategy”. A Consultation Document for the Government of the United Kingdom. 2003. Available online at: http://www.dcsf.gov.uk/consultations/downloadableDocs/towards a unified e-learning strategy.pdf

[DFES 05] Secretary of State for Education and Skills. “Department for Education and Skills: Five Year Strategy for Children and Learners”. A Consultation Document for the Government of the United Kingdom. 2005. Available online at: http://www.dfes.gov.uk/publications/5yearstrategy/

[Desire] Development of a European Service for Information on Research and Education (Desire), a collaboration between ten institutions from four European countries: Netherlands, Norway, Sweden and the UK. Available online at http://www.desire.org

[Diligenti et al. 00] Diligenti, M., Coetzee, F., Lawrence, S., Giles, C.L., Gori, M. “Focused Crawling using Context Graphs”. In Proceedings of 26th International Conference on Very Large Databases, VLDB 2000, Cairo, Egypt, pp. 527-534. September 10-14, 2000.

[Dill et al. 03] Dill, S., Eiron, N., Gibson, D., Gruhl, D., Guha, R., Jhingran, A., Kanungo, T., Rajagopalan, S., Tomkins, A., Tomlin, J., Zien, J. “SemTag and Seeker: Bootstrapping the Semantic Web via Automated Semantic Annotation”. In the Proceedings of the Twelfth International Conference on World Wide Web, WWW2003, Budapest, Hungary, pp. 178-186. 20th-24th May, 2003.

[Dogpile] Dogpile combines all the leading search engines in one results listing. Available online at http://www.dogpile.com

[DSpace] DSpace open source repository solution enables open sharing of a variety of digital content. Available online at http://www.dspace.org/

[Dripps et al. 06] Dripps, D., Casey, J., Proven, J. “After the Deluge: Navigating IPR Policy in Teaching and Learning Materials”. TrustDR Project Policy Report. September, 2006.

[Dripps et al. 07] Dripps, D., Casey, J., Proven, J. “Doing the Right Thing: Sources of Guidance for Good Practice with Metadata in Repositories”. TrustDR Project Report. January, 2007.

[Dublin Core] The Dublin Core Metadata Initiative is an open forum engaged in the development of interoperable online metadata standards that support a broad range of purposes and business models, available online at http://dublincore.org/

[Duval 01] Duval, E. “Standardized Metadata for Education: A Status Report”. In the Proceedings of the AACE World Conference on Educational Multimedia, Hypermedia and Telecommunications, Ed-Media 2001, C. Montgomerie, & V. Jarmo (Eds), pp. 458-463, Tampere, Finland. June 25th-30th, 2001.

[Duval et al. 02] Duval, E., Wason, T., Hodgins, W. "IEEE Standard for Learning Object Metadata". IEEE Standards Document 1484.12.1, produced for the Learning Technology Standards Committee. June, 2002.

[Duval & Hodgins 03] Duval, E. & Hodgins, W. "A LOM Research Agenda". In the Proceedings of the Twelfth International World Wide Web Conference, WWW2003, Budapest, Hungary. 20th-24th May, 2003.

[EdShare] EdShare supports collaboration and the sharing of educational materials across the University of Southampton. Available online at http://www.edshare.soton.ac.uk

[Edwards & Hardman 89] Edwards, D. & Hardman, L. “'Lost in Hyperspace': Cognitive Mapping and Navigation in a Hypertext Environment”. In Hypertext: Theory into Practice, R. McAleese (ed.), pp. 105-125, Intellect, Oxford, UK. 1989.

[EFC 05] Education for Change. “Regional Distributed e-Learning Baseline Study”. Higher Education Funding Council for England Report. June, 2005.

[Egghe & Rousseau 90] Egghe, L. & Rousseau, R. “Introduction to Informetrics”. Elsevier, 1990. Available online at http://uhdspace.uhasselt.be/dspace/handle/1942/587

[Eklund, 95] Eklund, J. “Cognitive Models for Structuring Hypermedia and Implications for Learning from the World Wide Web”. In Proceedings of the AusWeb 95 Conference, Ballina, NSW, Australia. pp. 111-116, 30 April – 2 May, 1995.

[Elgg] Elgg is an open, flexible social networking engine, designed to run any socially-aware application. Available online at http://www.elgg.org

[Eichmann et al. 94] Eichmann, D., McGregor, T., Dudley, D. "The RBSE Spider – Balancing Effective Search Against Web Load". In Proceedings of the First International World-Wide Web Conference, CERN, Geneva, Switzerland. O. Nierstarsz (ed.), pp 113-120. May 25-27, 1994.

[EPrints] EPrints open source software is a flexible platform for building high quality, high value repositories. Available online at http://www.eprints.org/

[Eurekster] Eurekster a social search tool which allows the creation of custom search portals powered by online communities. Available online at: http://www.eurekster.com

[Facebook] Facebook is a social utility that connects people with friends and others who work, study and live around them. Available online at http://www.facebook.com

[Farrell et al. 04] Farrell, R., Liburd, S., Thomas, J. "Dynamic Assembly of Learning Objects". In the Proceedings of the Thirteenth ACM International Conference on the World Wide Web, WWW2004, pp. 162–169, Manhattan, NY, USA. May 17th-20th, 2004.

[Fedora] Fedora is a flexible and extensible Digital Object and Repository Architecture. Available online at http://www.fedora-commons.org

[Fleming & Mills 92] Fleming, N. & Mills, C. "Helping Students Understand How They Learn". The Teaching Professor, vol. 7(4), Magma Publications: WI, U.S.A. 1992.

[Foster & Gibbons 05] Foster, N.F., & Gibbons, S. "Understanding Faculty to Improve Content Recruitment for Institutional Repositories". In D-Lib Magazine, vol. 11(1). January, 2005.

[Franklin & van Harmelen 07] Franklin, T. & van Harmelen, M. "Web 2.0 for Content for Learning and Teaching in Higher Education". JISC funded study. May 28th, 2007. Available online at http://www.jisc.ac.uk/publications/publications/web2andpolicyreport.aspx

[Freemind] Freemind is an open-source, Java Mind mapping tool. Available online at http://freemind.sourceforge.net

[Gadd et al. 03] Gadd, E., Oppenheim, C., Probets, S. "RoMEO Studies 1: The Impact of Copyright Ownership on Academic Author Self-Archiving". In the Journal of Documentation, vol. 59(3), pp. 243-277. 2003.

[Gagné 85] Gagné, R. “The Conditions of Learning (4th ed.)”. New York: Holt, Rinehart & Winston. 1985.

[Garfield 72] Garfield, E. “Citation Analysis as a Tool in Journal Evaluation”. In Science, vol.178, pp. 471-479, 1972.

[Gasparetti & Micarelli 03] Gasparetti, F., & Micarelli, A. “Adaptive Web Search Based on a Colony of Cooperative Distributed Agents”. In Cooperative Information Agents, vol. 2782, M. Klusch, S. Ossowski, A. Omicini, H. Laamanen (eds.), pp. 168-183, Springer-Verlag, 2003.

[Gasparetti & Micarelli 04] Gasparetti, F., & Micarelli, A. “Swarm Intelligence: Agents for Adaptive Web Search”. In Proceedings of the 16th European Conference on Artificial Intelligence, ECAI 2004, pp. 1019-1020, Valencia, Spain. August 22nd – 27th, 2004.

[Gatial & Balogh 06] Gatial, E. & Balogh, Z. “Identifying, Retrieving and Determining Relevance of Heterogeneous Internet Resources”. In Tools for Acquisition, Organisation and Presenting of Information and Knowledge, P.Navrat et al. (Eds.), Vydavatelstvo STU, Bratislava, pp. 15-21. Workshop, Nizke Tatry, Slovakia. 26th-28th September, 2006.

[Gatial et al. 07] Gatial, E., Balogh, Z., Hluchy, L., Vojtek, P. “Identification and Acquisition of Domain Dependent Internet Resources”. In the Proceedings of Informatics and Information Technologies, part 2, Tools for Acquisition, Organisation and Presenting of Information and Knowledge, pp. 68-78. Košice, Vydavateľstvo STU, Bratislava. 2007.

[GFS] Google File System is a scalable distributed file system for large distributed data-intensive applications. http://labs.google.com/papers/gfs.html

[Gillmore 98] Gillmore, D. “Small Portals Prove that Size Matters”. Technology column in San Jose Mercury News, 6th December, 1998. Available online at http://www.cse.iitb.ac.in/

[Gliffy] Gliffy is a free web-based diagram editor. Available online at http://www.gliffy.com

[GNOME] The GNOME project provides a desktop environment and development platform for Linux and Unix. Available online at http://www.gnome.org/.

[GNU] The GNU project has developed several Open Source licenses for the protection of freely available software. Available online at http://www.gnu.org/licenses/licenses.html

[Google] A search engine developed at Stanford University by Sergey Brin and Larry Page. Available online at http://www.google.com/

[Google API] Google SOAP Search API. Available online at: http://code.google.com/apis/soapsearch/reference.html

[Google Docs] Free web-based word processor and spreadsheet software for sharing and collaboration. Available online at http://docs.google.com

[GPL] GNU General Public License is a software copyright license. Available online at: http://www.gnu.org/copyleft/gpl.html

[Graf & Schnaider 97] Graf, F. & Schnaider, M. “IDEALS MTS - EIN modulares Training System für die Zukunft”. In the Proceedings of 8th Arbeitstreffen der GI-Fachgruppe 1.1.5/7.0.1 Intelligent Lehr-/Lernsysteme, Duisburg, C. Herzog (ed.), Technische Universität München, München. pp. 1-12, September 18-19. 1997.

[Gray 95] Gray, M. “Measuring the Growth of the Web”. Technical Report, Massachusetts Institute of Technology, 1995. Available online at: http://www.mit.edu/people/mkgray/growth/

[Green & Rubin 71] Greene, B. & Rubin, G. “Automated Grammatical Tagging of English”. Department of Linguistics, Brown University, Providence, RI, USA. 1971.

[Greeno et al. 96] Greeno, J.G., Collins, A.M., Resnick, L. “Cognition and Learning”. In the Handbook of Educational Psychology, First Edition, Simon & Schuster MacMillan: New York. 1996.

[Guthrie et al. 08] Guthrie, K., Griffiths, R., Maron, N. "Sustainability and Revenue Models for Online Academic Resources". An Ithaka Report produced for the Strategic Content Alliance. May, 2008.

[Hadoop] Apache Hadoop is a Free Java software framework that supports data intensive distributed applications running on large clusters of commodity computers. http://hadoop.apache.org/core/

[Harel 88] Harel, I. “Software Design for Learning: Children's Constructions of Meanings for Fractions and Logo Programming”. Doctoral Dissertation. MIT laboratory, Cambridge MA, USA. 1988.

[Harley et al. 06] Harley, D., Henke, J., Lawrence, S., Miller, I., Perciali, I., Nasatir, D., Kaskiris, C., Bautista, C. "Use and Users of Digital Resources: A Focus on Undergraduate Education in the Humanities and Social Sciences". Digital Resource Study Final Report. April, 2006.

[Harrsch 03] Harrsch, M. “RSS: The Next Killer App for Education”. In The Technology Source. July/August, 2003. Available online at http://technologysource.org/article/rss/

[Hewling 06] Hewling, A. “PROWE: Understanding the OU User Perspective”. A Report for the PROWE, Personal Repositories Online Wiki Environment, Project. September, 2006.

[Harman 92] Harman, D. “The DARPA TIPSTER Project”. In SIGIR Forum, vol. 26(2), pp. 26-28. 1992.

[Harvest] A distributed search engine framework developed at the University of Arizona. Available online at: http://sourceforge.net/projects/harvest/

[Hawking et al. 04] Hawking, D., Upstill, T., Craswell, N. “Towards Better Weighting of Anchors”. In Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Posters, pp. 512-513, The University of Sheffield, England. July 25th – 29th, 2004.

[Henze & Nejdl 00] Henze, N. and Nejdl, W. “Extendible Adaptive Hypermedia Courseware: Integrating Different Courses and Web Material”. In the Proceedings of the International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems, AH2000, pp. 109-120, Berlin: Springer-Verlag, Trento, Italy. August 28th-30th, 2000.

[Henze & Nejdl 01] Henze, N., Nejdl, W. “Adaptation in Open Corpus Hypermedia”. In the International Journal of Artificial Intelligence in Education, Special Issue on Adaptive and Intelligent Web-Based Systems, vol. 12, pp. 325–350. 2001.

[Heritrix] Heritrix is the Internet Archive's open-source, extensible, web-scale, archival-quality web crawler project. Available online at: http://crawler.archive.org/

[Hersovicia et al. 98] Hersovicia, M., Jacovia, M., Maareka, Y.S., Pellegb, D., Shtalhaima, M., Ura, S. “The Shark-Search Algorithm – An Application: Tailored Website Mapping”. In Proceedings of the Seventh World Wide Web Conference, WWW7, Brisbane, Australia, pp. 317-326. April 14-18, 1998.

[Hluchy et al. 07] Hluchy, L., Šeleng, M., Oravec, V., Budinska, I., Laclavik, M., Gatial, E., Balogh, Z., Ciglan, M. "Data Transition Chain". In the Proceedings of Informatics and Information Technologies, part 2, Tools for Acquisition, Organisation and Presenting of Information and Knowledge, pp. 79-91. Košice, Vydavateľstvo STU, Bratislava. 2007.

[Hölscher & Strube 00] Hölscher, C. & Strube, G. "Web Search Behavior of Internet Experts and Newbies". In Proceedings of the 9th International Conference on the World Wide Web, WWW9, Amsterdam, The Netherlands. pp 337-346, 2000.

[ht://Dig] ht://Dig is an open source indexing and search system. Available online at: http://www.htdig.org/

[IEEECNX] IEEE SPS Connexions Project for the sharing of signal processing content. Available online at http://www.ieeecnx.org

[iGoogle] iGoogle Personal Homepage. Available online at http://www.google.ie/ig

[IMS CP] IMS Global Learning Consortium Content Packaging is a standardised set of structures that can be used to exchange content. Available online at http://www.imsglobal.org/content/packaging/

[IMS LRM] IMS Information Model on using IEEE LOM, Learning Resource Metadata. Available online at http://www.imsglobal.org

[Internet Archive] The Internet Archive is a non-profit organisation, founded to build an Internet library, to offer permanent access to historical collections that exist in digital format. Available online at: http://www.archive.org

[ISC] The Internet Systems Consortium produces bi-annual statistics on the number of internet hosts on the WWW. Available online at http://www.isc.org/index.pl?/ops/ds/

[IWS] Internet World States publish statistics on the number of individual users of the WWW. Available online at http://www.internetworldstats.com/emarketing.htm

[Jochems et al. 04] Jochems, W., van Merriënboer, J., Koper, R. "Integrated E-Learning, Implications for Pedagogy, Technology and Organization". Routledge: New York, NY, USA. 2004.

[Jorum] Jorum, UK Higher Education Institutions Digital Repository. Available online at http://www.jorum.co.uk

[JORUM 04] "JORUM Scoping and Technical Appraisal Study: Volume III". Available online at: http://www.jorum.ac.uk/aboutus/archive/docs/vol3_Fin.pdf

[JTCL] The Java Text Categorizing Library is a Java implementation of libTextCat, a library for written language identification. Available online at: http://textcat.sourceforge.net/

[Kaelbling et al. 96] Kaelbling, L.P., Littman, M.L., Moore, A.W. "Reinforcement Learning: A Survey". In Journal of Artificial Intelligence Research, vol. 4, pp. 237-285, May, 1996.

[Kahn & O'Rourke 05] Kahn, P. & O'Rourke, K. "Understanding Enquiry-Based Learning". In the Handbook of Enquiry and Problem-based Learning: Irish Case Studies and International Perspectives, T. Barrett, I. Mac Labhrainn, H. Fallon (Eds), Galway: AISHE and CELT. 2005.

[Karamuftuoglu et al. 02] Karamuftuoglu, M., Jones, S., Robertson, S., Venuti, F., Wang, X.K. "Challenges Posed by Web-based Retrieval of Scientific Papers: Okapi Participation in TIPS". In the Journal of Information Science, vol. 28(1), pp. 3-17. 2002.

[Karbasi & Boughanem 06] Karbasi, S. & Boughanem, M. "Document Length Normalization Using Effective Level of Term Frequency in Large Collections". In the Proceedings of the 28th European Conference on Information Retrieval, ECIR 2006, London, England. LNCS 3936, pp. 72-83, Springer: Berlin. 10th-12th April, 2006.

[KartOO] KartOO is a visual meta search engine. Available online at: http://www.kartoo.com

[Kent 64] Kent, Allen. "Textbook on Mechanized Information Retrieval". Mathematics of Computation, Vol. 18, No. 88, pp. 686, Oct. 1964.

[Kirkwood 98] Kirkwood, A. "New Media Mania: Can Information and Communication technologies enhance the quality of open and distance learning?". In the International Journal on Distance Education, vol.19(2), pp.228-241. 1998.

[KiWi] The KiWi, or Knowledge in a Wiki, project combines the wiki philosophy with techniques from the Semantic Web community. Available online at http://www.kiwi-project.eu/

[Klein & Simmons 63] Klein, S. & Simmons, R.F. "A Computational Approach to Grammatical Coding of English Words". In Journal of the ACM, JACM, vol. 10, pp. 334-347. 1963.

[Kleinberg 99a] Kleinberg, J. M. "Hubs, Authorities and Communities". In ACM Computing Surveys, vol. 31(4es), December 1999.

[Kleinberg 99b] Kleinberg, J. M. "Authoritative Sources in a Hyperlinked Environment". In Journal of the ACM, vol. 46(5), pp 604-632, September 1999. Available online at: http://www.cs.cornell.edu/home/kleinber/auth.pdf

[Knallgrau] Knallgrau New Media Solutions. Available online at: http://www.knallgrau.at/en/company

[Knowles 80] Knowles, M.S. "The Modern Practice of Adult Education". The Adult Education Company: New York, Cambridge. 1980.

[KnowLib] Knowledge Discovery and Digital Library Research Group (KnowLib), The Department of Information Technology, Lund University, Sweden. Available online at http://www.it.lth.se/knowlib/

[Kobayashi & Takeda 00] Kobayashi, M. & Takeda, K. "Information Retrieval on the Web". In ACM Computing Surveys, vol. 32(2), pp. 144-173, ACM. June, 2000.

[Kohlschütter & Nejdl 08] Kohlschütter, C. & Nejdl, W. "A Densitometric Approach to Web Page Segmentation". In the Proceedings of the 17th ACM Conference on Information and Knowledge Management, CIKM 08, pp. 1173-1182, Napa Valley, California, USA. October 26th-30th, 2008.

[Kolb 84] Kolb, D. "Experiential Learning: Experience as the Source of Learning and Development". Prentice-Hall: NJ, U.S.A. 1984.

[Koper 03] Koper, E. J. R. "Combining reusable learning resources and services to pedagogical purposeful units of learning". In Reusing Online Resources: A Sustainable Approach to TEL, A. Littlejohn (Ed.), pp. 46-59, London: Kogan Page. 2003.

[Krovetz 93] Krovetz, R. "Viewing Morphology as an Inference Process". In Proceedings of the 16th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, ACM, New York, pp 191-202. 1993.

[Kulathuramaiyer 07] Kulathuramaiyer, N. "Mashups: Emerging Application Development Paradigm for a Digital Journal". In the Journal of Universal Computer Science, J.UCS, vol. 13(4), pp. 531-542. April, 2007.

[Labyrinth] Labyrinth is a lightweight mind mapping tool for the GNOME desktop. Available online at http://www.gnome.org/

[Lafferty & Zhai 01] Lafferty, J. & Zhai, C. “Document Language Models, Query Models, and Risk Minimization for Information Retrieval”. In the Proceedings of the 24th International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 111-119, New Orleans, Louisiana, USA, September 9th-13th, 2001.

[Lage et al. 04] Lage, J.P., Da Silva, A.S., Golgher, P.B., Laender, A.H.F. “Automatic Generation of Agents for Collecting Hidden Web Pages for Data Extraction”. In Journal of Data and Knowledge Engineering, vol. 49(2), pp. 177-196, 2004.

[Lane 06] Lane, A. "From Pillar to Post: Exploring the Issues Involved in Re-purposing Distance Learning Materials for use as Open Educational Resources". Working paper for the Open University. December, 2006.

[Laurillard 93] Laurillard, D. “Rethinking University Teaching: A Framework for the Effective Use of Educational Technology”, Routledge & Kegen, 1993.

[Laurillard 99] Laurillard, D. "New Technologies, Students and the Curriculum: The Impact of C&IT on Higher Education", in Higher Education Re-formed, P. Scott (ed), SRHE Publications, 1999.

[Laurillard 01] Laurillard D. “Rethinking University Teaching: A Framework for the Effective Use of Educational Technology”, 2nd edn. Routledge, London, 2001.

[Laurillard 07] Laurillard, D. “Technology, Pedagogy and Education: Concluding Comments”. In Technology, Pedagogy and Education, Vol. 16 (3), pp. 357-360, Routledge, October, 2007.

[Lave & Wenger 91] Lave, J. & Wenger, E. “Situated Learning: Legitimate Peripheral Participation”. Cambridge University Press: Cambridge, UK. 1991.

[Lawless et al. 05] Lawless, S., Wade, V., Conlan, O. “*Dynamic Contextual TEL - Dynamic Content Discovery, Capture and Learning Object Generation from Open Corpus Sources*”, In the Proceedings of the World Conference on E-Learning in Corporate, Government, Healthcare and Higher Education, E-Learn 2005, Vancouver, B.C., G. Richards (ed.), AACE, pp. 2158 – 2165. October 24th-28th, 2005.

[Lawless & Wade 06] Lawless, S. & Wade, V. "*Dynamic Content Discovery, Harvesting and Delivery, from Open Corpus Sources, for Adaptive Systems*", In the Proceedings of the Fourth International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems, AH 2006, V. Wade, H. Ashman, B. Smyth (eds.), Dublin, Ireland, LNCS 4018, Springer-Verlag, pp. 445–451. June 20th-23rd, 2006.

[Lawless et al. 06] Lawless, S., Dagger, D., Wade, V. "Towards Requirements for the Dynamic Sourcing of Open Corpus Learning Content", In the Proceedings of the World Conference on E-Learning in Corporate, Government, Healthcare, and Higher Education, E-Learn 2006, Honolulu, Hawaii, USA, T.C. Reeves & S.F. Yamashita (eds.). October 13th-17th, 2006.

[Lawless 07] Lawless, S. "Open Corpus Learning Content; Harvesting Knowledge to provide Equitable Access to Education for All". In the Proceedings of the International Student Conference Education Without Borders, EWB 2007, Abu Dhabi, United Arab Emirates. 25th-27th February, 2007.

[Lawless et al. 08a] Lawless, S., Hederman, L., Wade, V. "OCCS: Enabling the Dynamic Discovery, Harvesting and Delivery of Educational Content from Open Corpus Sources". In the Proceedings of the Eighth IEEE International Conference on Advanced Learning Technologies, I-CALT 2008, Santander, Spain. 1st-5th July, 2008.

[Lawless et al. 08b] Lawless, S., Hederman, L., Wade, V. "Enhancing Access to Open Corpus Educational Content: Learning in the Wild". In the Proceedings of the 21st ACM International Conference on Hypertext and Hypermedia, Hypertext 2008, Pittsburgh, PA, USA. 19th-21st June, 2008.

[Leiberman et al. 01] Leiberman, H., Fry, C., Weitzman, L. "Exploring the Web with Reconnaissance Agents". In Communications of the ACM, vol. 44(8), pp. 69-75. August, 2001.

[Lemur] The Lemur Toolkit is an open-source suite of tools designed to facilitate research in language modeling and information retrieval. Available online at: http://www.lemurproject.org

[LETSI] LETSI is an international, non-profit organisation dedicated to improving individual and organisational learning and performance. Available online at http://www.letsi.org

[libxml2] Libxml2 is the XML C parser and toolkit developed for the GNOME project. Available online at: http://xmlsoft.org

[Likert 32] Likert, R. “A Technique for the Measurement of Attitudes”. In the Archives of Psychology Journal, vol. 22(140), pp. 1-55. 1932.

[Link Buying] Official Google Webmaster Blog. “Information about buying and selling links that pass PageRank”. Available online at: http://googlewebmastercentral.blogspot.com/2007/12/information-about-buying-and-selling.html

[LinkedIn] LinkedIn is a business-oriented social networking site for creating networks of trusted contacts. Available online at http://www.linkedin.com

[Littlejohn 05] Littlejohn, A. “Community Dimensions of Learning Object Repositories”. In the Proceedings of the 22nd Annual Conference of the Australasian Society for Computers in Learning in Tertiary Education, H. Gross (Ed), pp. 3, Queensland University of Technology, Brisbane. 2005.

[LGPL] GNU Lesser General Public License is a software copyright license. Available online at: http://www.gnu.org/licenses/lgpl.html

[LOM] IEEE Learning Object Metadata. Available online at http://ltsc.ieee.org/wg12/

[Lovins 68] Lovins, J. “Development of a Stemming Algorithm”. In Mechanical Translation and Computational Linguistics, vol. 11, pp. 22-31, March, 1968.

[LTSN 02] LTSN Generic Centre. Circular 3: e-Learning. York: LTSN Generic Centre.

[Lucene] Apache Lucene is a full-featured text search engine library written entirely in Java. Available online at: http://lucene.apache.org/java/docs/index.html

[Lycos] A search engine from Lycos Inc. Available online at http://www.lycos.com

[Manning & Schütze 99] Manning, C. & Schütze, H. “Foundations of Statistical Natural Language Processing”. MIT Press. 1999

[Manning et al. 08] Manning, C., Raghavan, P., Schütze, H. “Introduction to Information Retrieval”. Cambridge University Press. 2008.

[Mapreduce] MapReduce is a programming model and an associated implementation for processing and generating large data sets. http://labs.google.com/papers/mapreduce.html

[Marchionini 95] Marchionini, G. “The Costs of Educational Technology: A Framework For Assessing Change”. In Proceedings of Ed-Media 95, World Conference of Educational Multimedia and Hypermedia, H. Maurer (Ed.), Graz, Austria, 1995.

[Marchiori 97] Marchiori, M. “The Quest for Correct Information on the Web: Hyper Search Engines”. In Proceedings of the 6th International World Wide Web Conference, WWW6, Santa Clara, CA, U.S.A. April 7-11, 1997.

[Margaryan 06] Margaryan, A. "CD-LOR Deliverable 7: Report on Personal Resource Management Strategies". A JISC deliverable report for the CD-LOR project. September, 2006.

[Marques Pereira et al. 05] Marques Periera, R.A., Molinari, A., Pasi, G. “Contextual Weighted Representations and Indexing Models for the Retrieval of HTML Documents”. In Soft Computing, vol. 9, pp. 481-492. 2005.

[Mayes & de Freitas 04] Mayes, T. & de Freitas, S. “Review of E-Learning Theories, Frameworks and Models”, JISC e-Learning Models Desk Study, 2004.

[Mayes & de Freitas 07] Mayes T. and de Freitas S. “Learning and e-learning: The Role of Theory”. In Rethinking Pedagogy in the Digital Age, H. Beetham & R. Sharpe (eds), pp. 13–15. Routledge, Abingdon and London, 2007.

[McBryan 94] McBryan, O. “Genvl and WWWW: Tools for taming the Web”. In Proceedings of the First International World-Wide Web Conference, CERN, Geneva, Switzerland. O. Nierstarsz (ed.), pp 1-13. May 25-27, 1994.

[McCallum & Nigam 98] McCallum, A. and Nigam, K. “A comparison of event models for Naive Bayes text classification.” In the Workshop on Learning for Text Categorization, at the Fifteenth National Conference on Artificial Intelligence, AAAI-98, Madison, Wisconsin, USA. 26th-30th, July, 1998.

[Meire et al. 07] Meire, M., Ochoa, X., Duval, E. “SAmgI: Automatic Metadata Generation v2.0”. In the Proceedings of the World Conference on Educational Multimedia, Hypermedia and Telecommunications 2007, EdMedia07, pp. 1195-1204, 25th-29th June, 2007.

[Menczer 01] Menczer, F. “Links tell us about Lexical and Semantic Web Content”. Computer Science Technical Report, August, 2001. Available online at http://arxiv.org/PS_cache/cs/pdf/0108/0108004v1.pdf

[Menczer 04] Menczer, F. “Lexical and Semantic Clustering by Web Links”. In Journal of the American Society for Information Science and Technology, vol. 55(14), pp. 1261-1269, 2004.

[Menczer & Belew 00] Menczer, F., & Belew, R.K. “Adaptive Retrieval Agents: Internalizing Local Context and Scaling up to the Web”. In Machine Learning, vol. 39(2/3), pp. 203-242, 2000.

[Menczer et al. 01] Menczer, F., Pant, G., Srinivasan, P., Ruiz, M.E. “Evaluating Topic-driven Web Crawlers”. In Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 241-249. September, 2001.

[Merlot] Multimedia Educational Resource for Learning and Online Teaching. Available online at http://www.merlot.org

[Mesick 76] Mesick, S. “Individuality in Learning”. Jossey-Bass: San Francisco. 1976.

[Metacombine] MetaCombine was an Emory University project whose aim was to experiment with methods to combine digital library resources and services. Available online at: http://www.metacombine.org/

[Micarelli & Gasparetti 07] Micarelli, A., & Gasparetti, F. “Adaptive Focused Crawling”. In The Adaptive Web: Methods and Strategies of Web Personalisation, Lecture Notes in Computer Science, Vol. 4321, Berlin: Springer Verlag, pp. 231-262. 2007.

[Micarelli et al. 07] Micarelli, A., Sciarrone, F., Marinilli, M. “Web Document Modelling”. In The Adaptive Web: Methods and Strategies of Web Personalisation, Lecture Notes in Computer Science, Vol. 4321, Berlin: Springer Verlag, pp. 155-192. 2007.

[Microsoft Popfly] Microsoft Popfly provides a facility for the generation of mashups, web pages and applications. Available online at http://www.popfly.com

[MIME] Multipurpose Internet Mail Extensions is an internet standard that describes the content type of an object on the web. http://www.iana.org/assignments/media-types/

[Mizuuchi & Tajima 99] Mizuuchi, Y., Tajima, K. “Finding Context Paths for Web Pages”. In Proceedings of the 10th ACM Conference on Hypertext and Hypermedia, Hypertext 99, pp. 13-22, Darmstadt, Germany, 1999.

[Mohr et al. 2004] Mohr, G., Kimpton, M., Stack, M., Ranitovic, I. "Introduction to Heritrix, an archival quality web crawler". In Proceedings of the 4th International Web Archiving Workshop (IWAW'04), Bath, UK, September 16th, 2004.

[Molinari et al. 03] Molinari, A., Pasi, G., Marques Pereira, R.A. "An Indexing Model of HTML Documents". In Proceedings of the 2003 ACM Symposium on Applied Computing, SAC, pp. 834-840, Melbourne, FL, U.S.A. March 9th-12th, 2003.

[Mooers 50] Mooers, C. N. "The theory of digital handling of non-numerical information and its implications to machine economics". Boston, Zator Co. 1950.

[Mooers 51] Mooers, C. N. "Making information retrieval pay". Boston, Zator Co. 1951

[MVC] Model-View-Controller is an architecture pattern used in software development. Definition available online at: http://java.sun.com/blueprints/patterns/MVC.html

[Myers-Briggs 80] Myers-Briggs, I. "Gifts Differing". Consulting Psychology Press: Palo Alto, CA, U.S.A. 1980.

[My Yahoo!] Customisable Yahoo! Homepage. Available online at http://my.yahoo.com

[Najjar et al. 03] Najjar, J., Ternier, S. & Duval, E. "The Actual Use of Metadata in ARIADNE: An Empirical Analysis". In Proceedings of 3rd International ARIADNE Conference, E. Duval (Ed), Leuven, Belgium,pp. 1-6, October, 2003.

[Nalanda] Nalanda iVia Focused Crawler is a focused web crawler developed by the UC Riverside Libraries iVia project. Available online at http://ivia.ucr.edu/projects/Nalanda/

[Nazou] Nazou is a Slovakian Government-funded project focused on the development of tools for the acquisition, organisation and maintenance of knowledge. Available online at: http://nazou.fiit.stuba.sk/

[NCSA] The National Center for Supercomputing Applications, the developers of the first major WWW browser, named Mosaic. NCSA also maintained a human edited catalogue of the web called "What's New". A copy of the June 1993 page is available online at http://wp.netscape.com/home/whatsnew/whats_new_0693.html

[NDLR] National Digital Learning Repository. Available online at http://www.learningcontent.edu.ie

[NDP 01] National Development Plan 2002-2006. 2001. Available online at http://www.ndp.ie

[Nelson 81] Nelson, T. H. “Literary Machines: The report on, and of, Project Xanadu concerning word processing, electronic publishing, hypertext, thinkertoys, tomorrow's intellectual revolution, and certain other topics including knowledge, education and freedom”. Mindful Press, Sausalito, California, U.S.A. 1981.

[Net::Google] A Perl interface to the Google SOAP Search API. Available online at: http://search.cpan.org/dist/Net-Google/

[Net::OAI] Net::OAI::Harvester is a Perl package for harvesting metadata using OAI-PMH. Available online at: http://search.cpan.org/dist/OAI-Harvester/

[Netcraft] The Netcraft Web Server Survey which is available with detailed explanation at http://news.netcraft.com/archives/web_server_survey.html provides monthly statistics on the number of web servers on the WWW. Also available in archive form dating back to 1995 at http://survey.netcraft.com/Reports/

[Newell 90] Newell, A. “Unified Theories of Cognition”. Harvard University Press: Cambridge, MA, U.S.A. 1990.

[NISO 04] National Information Standards Organisation. “Understanding Metadata”. NISO Press. 2004. Available online at http://www.niso.org/publications/press/

[NIST] The National Institute of Standards and Technology is a U.S. federal technology agency that develops and promotes measurement, standards, and technology. Available online at: http://www.nist.gov

[Novak & Cañas 04] Novak, J. D., Cañas, A. J. “Building on New Constructivist Ideas and CmapTools to Create a New Model for Education”. In proceedings of the first International Conference on Concept Mapping, A. J. Cañas, J. D. Novak, F. M. González (Eds.), Pamplona, Spain, 2004.

[Ntoulas et al. 05] Ntoulas, A., Cho, J., Olston, C. “What’s New on the Web? The Evolution of the Web from a Search Engine Perspective”. In Proceedings of the Thirteenth International Conference on the World Wide Web, WWW2004, New York, NY, USA. May 17th-22nd, 2004.

[Ntoulas et al. 05] Ntoulas, A., Zerfos, P., Cho, J. “Downloading Textual Hidden Web Content through Keyword Queries”. In Proceedings of the ACM/IEEE Joint Conference on Digital Libraries, JCDL 05, pp. 100-109, Denver, CO, USA. June 7-11, 2005.

[Nutch] Nutch is an open source web-search solution based upon Lucene. Available online at: http://lucene.apache.org/nutch

[NutchWAX] Nutch and Web Archive eXtensions is a tool for indexing and searching web archive collections. Available online at: http://archive-access.sourceforge.net/projects/nutch/

[NWA] The Nordic Web Archive Toolset is a software package for accessing archived web documents. Available online at: http://nwa.nb.no/

[OAI] The Open Archives Initiative develops and promotes interoperability standards that aim to facilitate the efficient dissemination of content. Available online at: http://www.openarchives.org/

[OAI-PMH] The Open Archives Initiative Protocol for Metadata Harvesting (OAI-PMH) is a low-barrier mechanism for repository interoperability. http://www.openarchives.org/pmh/

[OCW] OpenCourseWare is a free, web-based publication service for MIT’s educational content. Available online at http://ocw.mit.edu

[ODP] The Open Directory Project is the most comprehensive human-reviewed directory of the web. Available online at: http://www.dmoz.org

[ODP Submission] ODP submission guidelines for the addition of a site to the directory. Available online at http://www.dmoz.org/add.html

[OER Commons] OER Commons provides access to, and descriptions of, freely available educational resources from around the World. Available online at http://www.oercommons.org/

[OpenLearn] OpenLearns LearningSpace provides free access to course materials from the Open University. Available online at http://openlearn.open.ac.uk/

[OpenGoo] OpenGoo is open source software designed to deliver a web-based suite of office tools. Available online at http://opengoo.org

[Omega] Omega is an open source search interface commonly used in conjunction with Xapian, Available online at: http://xapian.org/docs/omega/overview.html

[O'Reilly 07] O'Reilly, T. "What Is Web 2.0: Design Patterns and Business Models for the Next Generation of Software". In Communications and Strategies - International Journal of Digital Economics, vol. 65, pp. 17-37. 2007.

[Page et al. 98] Page, L., Brin, S., Motwani, R., Winograd, T. "The PageRank Citation Ranking: Bringing Order to the Web". Technical report, Stanford University, Stanford, CA, 1998. Available online at http://dbpubs.stanford.edu:8090/pub/1999-66

[Papert 93] Papert, S. "The Children's Machine - Rethinking School in the Age of the Computer". Basic Books: New York. 1993.

[Papert & Harel 91] Papert, S. & Harel, I. "Constructionism - Chapter 1: Situating Constructionism". Ablex Publishing Corporation: New York. 1991.

[Pasternack & Roth 09] Pasternack, J. and Roth, D. "Extracting Article Text from the Web with Maximum Subsequence Segmentation". In the Proceedings of the 18th International World Wide Web Conference, WWW2009, pp. 971-980, Madrid, Spain. April 20th-24th, 2009.

[Peal & Wilson 01] Peal, D. & Wilson, B. "*Activity Theory and Web-Based Training*". In Web-based training, B. Khan (Ed.), pp. 147-153, Educational Technology Publications: Englewood Cliffs, NJ, USA. 2001.

[Piaget & Inhelder 69] Piaget, J. & Inhelder, B. "The Psychology of the Child". Basic Books: New York. 1990.

[Pinkerton 00] Pinkerton, B. "WebCrawler: Finding What People Want". PhD Thesis submitted to the University of Washington. 2000.

[Pinski & Narin 76] Pinski, G. & Narin, F. "Citation Influence for Journal Aggregates of Scientific Publications: Theory, with Application to the Literature of Physics". In Information Processing and Management, vol.12, pp. 297-312, 1976.

[Pittard 04] Pittard, V. "Evidence for E-Learning Policy". In Technology, Pedagogy and Education, vol. 13(2), pp. 181-194. July, 2004.

[Pizzani & Billsus 07] Pazzani, M. and Billsus, D. "Content-based Recommendation Systems". In The Adaptive Web: Methods and Strategies of Web Personalisation, Lecture Notes in Computer Science, Vol. 4321, Berlin: Springer Verlag, pp. 325-341. 2007.

[Ponte & Croft 98] Ponte, J.M. & Croft, W.B. “A Language Modeling Approach to Information Retrieval”, In the proceedings of the 21st Annual ACM SIGIR Conference, pp. 275-281, Melbourne, Australia. August 24th-28th 1998.

[Porter 80] Porter, M.F. “An Algorithm for Suffix Stripping”. In Program, vol. 14(3), pp 130?137, 1980.

[Porter 04] Porter, J. “Home Alone? How Content Aggregators Change Navigation and Control of Content”. In Digital Web Magazine, November 3rd, 2004. Available online at http://www.digital-web.com/articles/home_alone_content_aggregators/

[Prensky 05] Prensky, M. “Listen to the Natives”. In the Journal of Educational Leadership, vol. 63(4), pp. 8-13. December, 2005.

[Pritchard 69] Pritchard, A. “Statistical Bibliography or Bibliometrics”. In Journal of Documentation, vol.25, pp. 348-349, 1969.

[ProgrammableWeb 08] The ProgrammableWeb site provides information on Mashups, APIs and Web 2.0. Available at http://www.programmableweb.com/mashups

[Putnam 95] Putnam, H. “Against the New Associationism”. In Speaking Minds: Interviews with Twenty Eminent Cognitive Scientists, P. Baumgartner and S. Payr (eds.), pp. 177-188. 1995.

[Quesenbery 03] Quesenbery, W. “Designing a Search People can really use”. In the Society for Technical Communication Intercom Magazine, pp. 18-21. December 2003.

[Rabinowitz 03] Rabinowitz, J. “How to Index Anything”. In Linux Journal Magazine, Vol. 111, July 2003. Pages 82-88.

[Radev et al. 05] Radev, D., Fan, W., Qi, H., Wu, H., Grewal, A. “Probabilistic Question Answering on the Web”. In the Journal of the American Society for Information Science and Technology, vol. 56(6), pp. 571–583. April, 2005.

[Raghavan & Garcia-Molina 01] Raghavan, S., Garcia-Molina, H. “Crawling the Hidden Web”. In Proceedings of the 27th International Conference on Very Large Data Bases, VLDB 01, Morgan Kaufmann Publishers Inc., pp.129-138, San Francisco, CA, USA. 2001.

[Rainbow] Rainbow is a program that performs statistical text categorisation. Available online at: http://www.cs.cmu.edu/

[Reenskaug 79] Reenskaug, T. “Models-Views-Controllers”. Technical report, XEROX Palo Alto Research Center, Palo Alto, CA.

[Rennie & McCallum 99] Rennie, J., & McCallum, A. “Using Reinforcement Learning to Spider the Web Efficiently”. In Proceedings of the Sixteenth International Conference on Machine Learning, ICML-99, Bled, Slovenia, pp. 335-343. June 27-30, 1999.

[Resnick & Resnick 92] Resnick, L.B. & Resnick, D.P. “Assessing the Thinking Curriculum: New Tools for Education Reform”. In Changing Assessment: Alternative Views of Aptitude, Achievement and Instruction, pp. 37-75, B.R. Gifford & M.C. O’Connor (eds.), Kluwer: Boston. 1992.

[Ring & MacLeod 01] Ring, J.L. & MacLeod, D. “The BELLE Project: Towards a National Digital-Content Repository”. In the Canadian Journal of Communication, vol. 26(3). 2001.

[Robertson & Sparck Jones 76] Robertson, S.E., Sparck Jones, K. “Relevance Weighting of Search Terms”. In Journal of the American Society for Information Science, vol. 27, pp. 129-146. 1976.

[Robots.txt] The Robot Exclusion Standard is an open standard for the definition of web crawler permissions and preferences for a website. The specification can be found online at: http://www.robotstxt.org/orig.html and the W3C recommendations can be found online at: http://www.w3.org/TR/html4/appendix/notes.html#h-B.4.1.1

[Roussinov et al. 08] Roussinov, D., Weiguo, F., Robles-Flores, J. “Beyond Keywords: Automatic Question Answering on the Web”. In the Communications of the ACM, vol. 51(9), pp. 60-65. September, 2008.

[Sakai] Sakai is a collaboration and community management platform. Available online at http://sakaiproject.org

[Salton 71] Salton, G. “The SMART Retrieval System – Experiments in Automatic Document Processing”. Prentice-Hall Inc., NJ, USA. 1971.

[Salton 75] Salton, G., Wong, A. and Yang, C.S. “A vector space model for automatic indexing”. In Communications of the ACM, 18 (11). 613-620. 1975.

[Salton 83] Salton, G., Fox, E.A. and Wu, H. “Extended Boolean information retrieval”. In Communications of the ACM, 26 (11). 1022-1036. 1983.

[Salton 89] Salton, G. “Automatic Text Processing: The Transformation, Analysis, and Retrieval of Information by Computer”. Addison-Wesley, 1989.

[Salton & Buckley 88] Salton, G. & Buckley, C. “Term Weighting Approaches in Automatic Text Retrieval”. In Information Processing and Management, vol. 24(5), pp. 513-523. 1988.

[Salton & McGill 84] Salton, G. & McGill, M.J. “Introduction to Modern Information Retrieval”. McGraw Hill International Book Company. 1984.

[Salton & Yang 73] Salton, G. & Yang, C.S. “On the Specification of Term Values in Automatic Indexing”. In Journal of Documentation, vol. 29(4), pp. 351-372. December 1973.

[Sampath 85] Sampath, G. “An Introduction to Text Processing: A Systematic Approach to the Study of Text Structure and Operations and the Design of Text Processing Software”. River Valley Publishing, Jeffersontown, KY, U.S.A. 1985.

[Schonfeld & Shivakumar 09] Schonfeld, U. & Shivakumar, N. “Sitemaps: Above and Beyond the Crawl of Duty”. In the Proceedings of the 18th International World Wide Web Conference, WWW2009, pp. 991-1000, Madrid, Spain. April 20th-24th, 2009.

[SCORM] Sharable Content Object Reference Model, SCORM. Available online at http://www.adlnet.gov/Scorm/

[SCORM 06] "*SCORM 2004 3rd Edition – Sharable Content Object Reference Model: Overview*". Advanced Distributed Learning. November, 2006.

[Schneiderman et al. 97] Schneiderman, B., Byrd, D., Croft, W. B. “Clarifying Search: A User-Interface Framework for Text Searches”. In D-Lib Magazine. January, 1997.

[Scott 04] Scott, J. “Assessing the Relevance of the Review of e-learning Theories, Frameworks and Models and the Mapping Table to Designers”, JISC e-Learning Models Desk Study, 2004.

[Shneiderman 98] Shneiderman, B. “Designing the User Interface: Strategies for Effective Human-Computer Interaction”, Addison Wesley Longman, Inc., Third Edition. 1998.

[Siegel & Kirkley 97] Siegel, M. A., & Kirkley, S. “ Moving toward the digital learning environment: The future of Web-based instruction”. In Web-Based Instruction. B. H. Khan (Ed.), Educational Technology Publications, Englewood Cliffs, NJ. 1997.

[Skinner 75] Skinner, B. F. “About Behaviourism”, Cape, London. 1975.

[Skinner 77] Skinner, B. F. “Why I am not a Cognitive Psychologist”. In Behaviourism, vol. 5, pp. 1-10. 1977.

[Snap] Snap is a search interface with dynamic content preview. Available online at: http://www.snap/com

[SOAP] Simple Object Access Protocol is a lightweight protocol for the exchange of structured information in a decentralised, distributed environment. Available online at: http://www.w3.org/TR/soap/

[SOAP::Lite] SOAP::Lite for Perl is a collection of Perl modules which provides a simple and lightweight interface to the Simple Object Access Protocol. Available online at: http://www.soaplite.com/

[Sparck Jones 72] Sparck Jones, K. "A Statistical Interpretation of Term Specificity and its Application in Retrieval". In Journal of Documentation, vol. 28(1), pp. 11-21. March, 1972.

[Sparck Jones & Willet 97] Sparck Jones, K. & Willet, P. "Readings in Information Retrieval". San Francisco: Morgan Kaufmann, 1997.

[Sparck Jones et al. 00a] Sparck Jones, K., Walker, S., Robertson, S.E. "A Probabilistic Model of Information Retrieval: Development and Comparative Experiments – Part 1". In Information Processing and Management, vol. 36(6), pp. 779-808. 2000.

[Sparck Jones et al. 00b] Sparck Jones, K., Walker, S., Robertson, S.E. "A Probabilistic Model of Information Retrieval: Development and Comparative Experiments – Part 2". In Information Processing and Management, vol. 36(6), pp. 809-840. 2000.

[Spock] Spock, a people search engine. Available online at http://www.spock.com

[Stanley 06] Stanley, T. "Web 2.0: Addressing the Barriers to Implementation in a Library Context". QA Focus Briefing Document no. 103, UKLON. August, 2006.

[Stauffer, 96] Stauffer, K. "Student Modelling and Web-Based Learning Systems". Athabasca University, Canada. 1996. Available online at http://ccis.athabascau.ca/html/students/stupage/Project/initsm.htm

[Steichen et al. 09] Steichen, B., Lawless, S., O'Connor, A. & Wade, V. "Dynamic Hypertext Generation for Reusing Open Corpus Content". In the Proceedings of the 20th ACM Conference on Hypertext and Hypermedia, Hypertext 2009, Torino, Italy. 29th June – 1st July, 2009.

[SURT] Sort-friendly URI Reordering Transform, a transform applied to URIs which makes their left-to-right representation match the natural hierarchy of domain names. Definition available online at: http://www.crawler.archive.org/articles/user_manual/glossary.html#surt

[SURT Prefix] A shared prefix string of all SURT form URIs in the same subject area, for web crawling. Definition available online at: http://www.crawler.archive.org/articles/user_manual/glossary.html#surtprefix

[Swing] Swing is a widget toolkit for Java, aiding the development of graphical user interfaces. It is part of the Java Foundation Classes API from Sun. Available online at: http://java.sun.com/products/jfc/reference/faqs/index.html

[Swish-e] Simple Web Indexing System for Humans – Enhanced (Swish-e), a flexible and free open source system for indexing collections of Web pages. Available online at http://www.swish-e.org

[Tafiti] Tafiti is an experimental search interface from Microsoft powered by Silverlight. Available online at: http://www.tafiti.com/Original/default.aspx

[Technorati 08] Technorati Media. "The State of the Blogosphere 2008". Available online at http://www.technorati.com/blogging/state-of-the-blogosphere/

[Theng 97] Theng, Y. L. "Addressing the 'Lost in Hyperspace' Problem in Hypertext". PhD Thesis, Middlesex University, London. 1997.

[Tiropanis et al. 09] Tiropanis, T., Davis, H.C., Millard, D. & Weal, M. "Semantic Technologies for Learning and Teaching in the Web 2.0 Era: A Survey of UK Higher Education". In the Proceedings of the Web Science 2009 Conference, WebSci'09, Athens, Greece. 18th-20th March, 2009.

[TREC] Text REtrieval Conference – TREC. Available online at http://trec.nist.gov/

[TrueLocal] TrueLocal provides local transactional search in the U.S.A. Available online at http://www.truelocal.com

[Tuckey 92] Tuckey, C. "Uses of New Technology in Higher Education - Guiding Principles". In ICBL Reports, 001/92 Institute for Computer Based Learning, Heriot-Watt University, Edinburgh. 1992.

[Tullis & Stetson 04] Tullis, T.S. & Stetson, J.N. "A Comparison of Questionnaires for Assessing Website Usability", In the Proceedings of the Usability Professional Association Conference, UPA 2004, Minneapolis, Minnesota, USA. June 7th-11th, 2004.

[U-CREATe FAQ] A list of frequently asked questions regarding U-CREATe. Available online at https://www.cs.tcd.ie/

[UNESCO] UNESCO stands for the United Nations Educational, Scientific and Cultural Organization. Available online at http://www.unesco.org

[Urban-Lurain 96] Urban-Lurain, M. "Intelligent Tutoring Systems: An Historic Review. In the Context of the Development of Artificial Intelligence and Educational Psychology". 1996. Available online at http://www.cse.msu.edu/rgroups/cse101/ITS/its.htm

[van Rijsbergen 79] van Rijsbergen, K. "Information Retrieval". London, England, Butterworths & Co. Ltd. 1979.

[Verbert & Duval 04] Verbert, K. & Duval, E. "Towards a Global Architecture for Learning Objects: A Comparative Analysis of Learning Object Content Models". In the Proceedings of the World Conference on Educational Multimedia, Hypermedia and Telecommunications 2004, ED-MEDIA 04, Lugano, Switzerland, L. Cantoni & C. McLoughlin (Eds.), pp. 202-208. 2004.

[Vieira et al. 06] Vieira, K., da Silva, A., Pinto, N., de Moura, E., Cavalcanti, J. & Freire, J. "A Fast and Robust Method for Web Page Template Detection and Removal". In the Proceedings of the 15th ACM International Conference on Information and Knowledge Management, CIKM 06, Arlington, Virginia, USA. November 6th-11th, 2006.

[VLib] WWW Virtual Library. The oldest human maintained catalogue of the web. Available online at http://vlib.org

[von Glasersfeld 91] von Glasersfeld, E. "Radical Constructivism in Mathematics Education". Editor. Kluwer Academic. 1991.

[von Glasersfeld 95] von Glasersfeld, E. "A Constructivist Approach to Teaching". In Constructivism in Education, L.P. Steffe & J. Gale (Eds), Lawrence Erlbaum Associates. 1995.

[Voorhees 99] Voorhees, E.M. "The TREC-8 Question Answering Track Report". In the Proceedings of the 8th Text Retrieval Conference, TREC-8, Gaithersburg, Maryland, USA, pp. 77-82. November 17th-19th, 1999.

[VUE] VUE is an open-source, Java concept mapping application. Available online at http://vue.tufts.edu/

[Vygotsky 34] Voygotsky, L.S. "*Thought and Language*". MIT Press: Cambridge, MA, USA.

[Vygotsky 78] Voygotsky, L.S. "*The Mind in Society: The Development of Higher Psychological Processes*". Harvard University Press: MA, USA.

[Wade & Ashman 07] Wade, V., & Ashman, H. "Evolving the Infrastructure for Technology-Enhanced Distance Learning". In IEEE Internet Computing Special Issue on Distance Learning, vol. 11(3), pp 16-18, June 2007.

[Wade & Power 98] Wade, V. and Power, C. "Network Based Delivery and Automated Management of Virtual University Courses". In Proceedings of the 10th World Conference on Educational Multimedia and Hypermedia & World Conference on Educational Telecommunications (ED-MEDIA & ED-TELCOM '98), pp 1421-1427, Freiburg, Germany, June, 1998.

[Wade et al. 04] Wade, V., Lee, M., McMullin, B., Mulholland, C., MacLabhrainn, I., Slowey, M., Fox, S., McKeown, C. "TEL as a Strategic Imperative for Universities in Ireland". Briefing Paper for Symposium on TEL as a Strategic Imperative for Universities in Ireland, Dublin City University. 4th November, 2004.

[Wayback] The Wayback Machine is a searchable digital internet archive, containing snapshots of websites over periods of time from Alexa Internet. Available online at: http://www.archive.org/web/web.php

[WebCrawler] A history of WebCrawler on Brian Pinkerton's personal site, available at http://thinkpink.com/bp/WebCrawler/History.html

[WebGlimpse] WebGlimpse is indexing and search software which runs on Unix. Available online at http://webglimpse.net/

[Weller et al., 03] Weller, M., Pegler, C., Mason, R. "Putting the pieces together: What Working with Learning Objects means for the Educator". In the Proceedings of the Second eLearnInternational World Summit, Edinburgh International Conference Centre, Edinburgh, Scotland. February 18th-19th, 2003.

[WERA] Web ARchive Access is a freely available solution for searching and navigating archived web document collections. Available online at: http://archive-access.sourceforge.net/projects/wera/

[White 06] White, D. "Spire Change Report". Project report for Spire, a JISC funded project to address policy, middleware and software issues to facilitate a community of authenticated peer-2-peer early adopters within UK HEIs. October, 2006.

[Wikipedia] A free encyclopedia built collaboratively using Wiki software. Available online at http://www.wikipedia.org

[Wikiversity] Wikiversity is a community devoted to collaborative learning. Available online at http://en.wikiversity.org

[Wiley 00] Wiley, D. A. "Learning Object Design and Sequencing Theory". PhD Thesis submitted to Brigham Young University. June, 2000. Available online at http://opencontent.org/docs/dissertation.pdf

[Wilson & Myers 00] Wilson, B.G. & Myers, K.M. "Situated Cognition in Theoretical and Practical Context". In Theoretical Foundations of Learning Environments, pp. 57-88, D.H. Jonassen & S.M. Land (eds.), Lawrence Erlbaum: New York. 2000.

[Wilson et al. 04] Wilson, S., Blinco, K., Rehak, D. "Service-Oriented Frameworks: Modelling the infrastructure for the next generation of e-Learning Systems". A Paper prepared on behalf of DEST (Australia), JISC-CETIS (UK), and Industry Canada, 2004.

[WSDL] The Web Services Description Language is an XML-based language that provides a model for describing Web services. Available online at: http://www.w3.org/TR/wsdl

[Xapian] Xapian is an open source indexing and search library, implemented in C++. Available online at http://xapian.org/

[Yahoo!] A search engine from Yahoo!. Available online at http://search.yahoo.com

[Yahoo! Answers] Social search platform where people can ask questions, post answers and browse popular threads. Available online at: http://answers.yahoo.com

[Yahoo! Kids] Search portal including aspects of social search aimed at Children. Available online at: http://kids.yahoo.com

[Yahoo! Pipes] Yahoo! Pipes is a composition tool for the creation of mashups from web content. Available online at http://pipes.yahoo.com

[Yang 01] Yang, K. "Combining Text and Link-Based Retrieval Methods for Web-IR". In Proceedings of the 9th Text REtrieval Conference, TREC 9, pp. 609-618. 2001.

[Yang & Maglaughlin 00] Yang, K. & Maglaughlin, K.L. “IRIS at TREC-8”. In Proceedings of the 8th Text REtrieval Conference, TREC 8, pp. 645-656. 2000.

[YourLocal] YourLocal provides local transactional search within Ireland. Available online at http://www.yourlocal.ie

[Zastrocky et al. 06] Zastrocky, M., Harris, M., Lowendahl, J-M. “Hype Cycle for Higher Education, 2006”. Gartner Report G00139174. 30th June, 2006.

[ZDNet] ZDNet Interview with Geoff Johnson, research vice president at Gartner Group. Available online at http://news.zdnet.co.uk/hardware/0,1000000091,39413759,00.htm

[Zipf 35] Zipf, G.K. “Psycho-Biology of Languages”. Houghton-Mifflin, 1935.

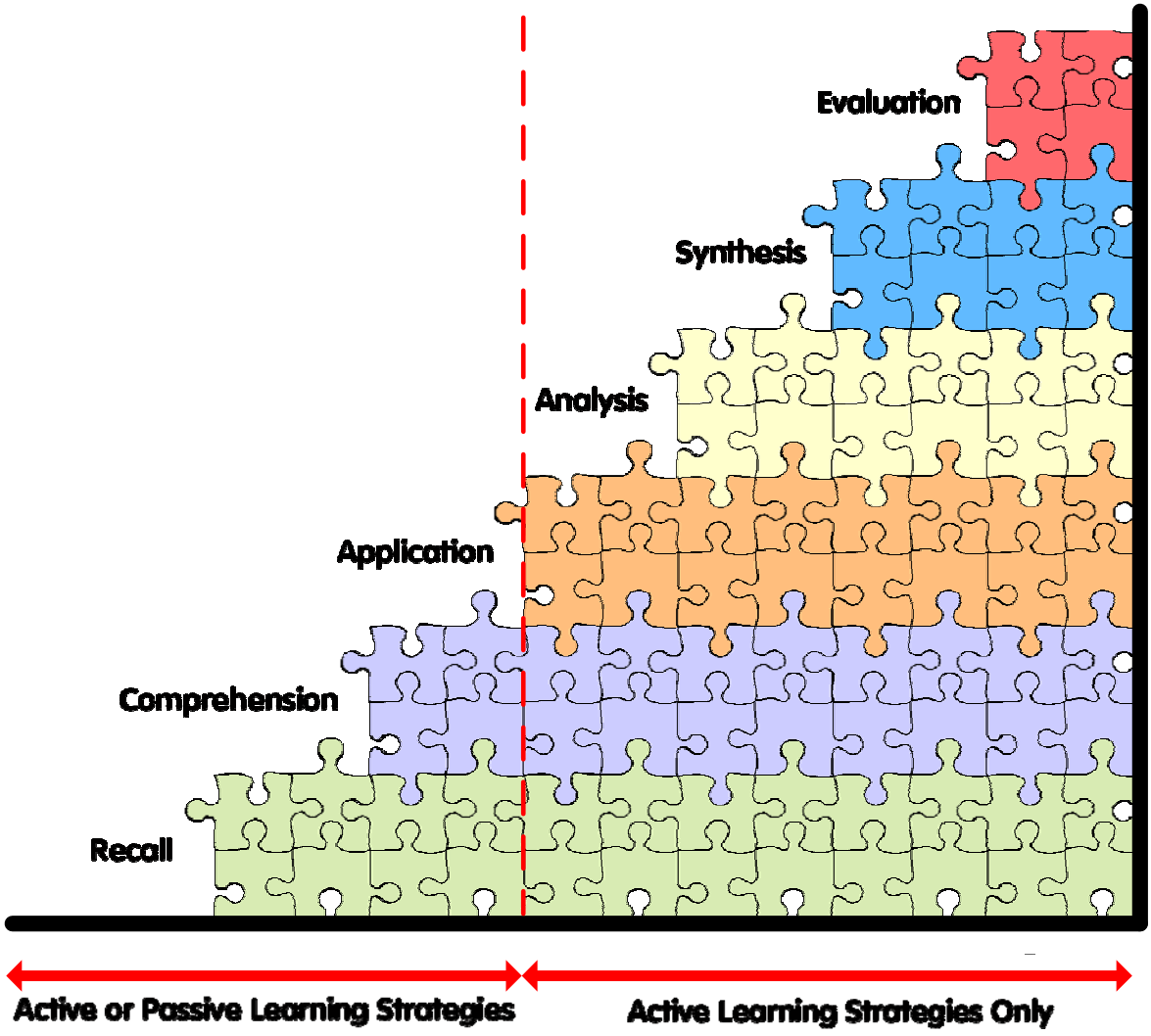
[Zipf 49] Zipf, G.K. “Human Behaviour and the Principle of Least Effort”. Addison-Wesley, 1949.

Appendices

Appendix A – Educational Theories and Approaches

Blooms Taxonomy

Blooms' taxonomy of educational objectives is a classification of the learning objectives and skills that educators can set for learners. The taxonomy earns its name from one of its proposers, Benjamin Bloom, who was an educational psychologist based at the University of Chicago. Bloom developed a taxonomy which describes six hierarchical levels of objectives, namely: Recall, Comprehension, Application, Analysis, Synthesis and Evaluation[9]. The hierarchical nature of the taxonomy means learning at the higher cognitive levels is dependent upon prerequisite knowledge and skills having been attained at lower levels.



Blooms Taxonomy of Educational Objectives

[9] In a recent revision of Bloom's Taxonomy [Anderson et al. 01] the six hierarchical levels of objectives have been renamed as: Remembering, Understanding, Applying, Analysing, Evaluating and Creating.

Recall is defined as the learner's ability to acquire and subsequently recall information which they have processed. This forms the lowest level of the hierarchy. An example of Recall would be a learner's ability to learn and then accurately recite a poem or the ability to accurately replicate a mathematical formula.

Comprehension is defined as the learner's ability to apply meaning to the processed information. Comprehension commonly occurs in conjunction with Recall and in certain cases Recall can be a prerequisite of Comprehension. An example of Comprehension would be a learner's ability to understand and verbalise the meaning of a line in a poem or the ability to paraphrase the notation of a mathematical formula.

Application is defined as the ability to use acquired knowledge in new or real-life situations. This relates to the application of rules, laws, methods and theories. An example of Application would be a learner identifying an example of a metaphor or adjective in a poem or the ability to instances of triangle types outside of the mathematical textbook.

Analysis is defined as the ability to decompose complex information into individual concepts and understand the relationships between each concept. An example of Analysis would be the ability to identify the poetic form used in a poem or the ability to determine the strategies necessary to solve a mathematical problem.

Synthesis is defined as the ability to combine individual concepts or pieces of information to create something novel. An example of Synthesis would be a learner writing a new poem into a particular poetic form or combining several different formulae to solve a mathematical problem.

Evaluation is defined as the ability to perform a comparison based on cognitive standards which have been developed through the lower levels of the taxonomy. An example of Evaluation would be a learner's ability to analyse a peer's poem based on the principles of poetic form or the ability to determine the efficiency of two separate methods of solving a mathematical problem.

Constructivism and Constructionism

Constructivism [Piaget & Inhelder 69], while building upon cognitivist approaches, has subtle differences. In cognitivism, the educator retains control over the educational process and remains responsible for directing the learners thinking. In constructivism the educator must accept the autonomy of the learner and act as a facilitator of knowledge acquisition rather than merely a disseminator of information [Carlile & Jordan 05]. The goal of the educator in constructivism is to engage the learner in active participation, problem solving, interdisciplinary work, reflection and discussion.

One of the key principles of constructivism is its emphasis on diversity in learning. Research has shown that adults construct knowledge in different ways to children [Knowles 80]. Though Piaget's research focused mainly on the psychology of children, the principles of this approach are also applicable to adult learners. Some researchers also believe that learners can be categorised by learning style [Myers-Briggs 80] [Fleming & Mills 92] [Kolb 84]. These are described as characteristic modes of perceiving, remembering, thinking, problem solving and decision making [Mesick 76]. However, the true value of learning styles has since come under some scrutiny and their effectiveness is unclear [Brown et al. 07].

It is essential in constructivist learning that the educator understand the learners thinking and encourage them to reflect on their knowledge constructs as a means to improve them. This can be achieved through verbal class discussions or through knowledge visualization techniques such as Mindmapping. Social interaction can also be an important stimulus for this reflection and for motivating knowledge construction and adaptation [von Glasersfeld 91] [von Glasersfeld 95].

Constructionism is an educational method based upon constructivist approaches [Papert & Harel 91]. It shares constructivism's connotation of learning as "building knowledge structures". Constructionism builds upon the notions of constructivism by adding the idea that this formulation of knowledge structures occurs most prolifically in a context where the learner is "consciously engaged in constructing a public entity, whether it's a sand castle on the beach or a theory of the universe". This implies that individuals learn best when they are in the active roles of designer and constructor and are increasingly motivated if the item they are constructing will be seen, critiqued or used by their peers. Papert uses the diffusion of

cybernetic construction kits, such as Lego Mindstorms™, into the lives of children as an example of how constructionist methods could change the context of the learning of mathematics [Papert 93]. The construction and programming of such models will not only improve children's understanding of mathematical concepts, the children will also have more motivation to further learn math's if it enables them to build better, more complex mindstorm models.

One practical outcome of this theory of human learning is that the learning medium must create a situation where the learner has the freedom to exercise judgment about what is to be learned and at what pace. Papert states that the real benefit of computers in education will be seen when educational systems begin to shift the balance between the instructional transfer of knowledge *to* students and the production/construction of knowledge *by* students. In her PhD Thesis [Harel 88], Idel Harel documented experiments which showed that children's attention could be held for an hour a day for periods of several months by making, as opposed to using, educational software, even if the children consider the subject of the software to be boring in its classroom form.

Mind Mapping

Mind mapping is a technique used to generate, visualise, structure and classify knowledge. A diagram is generated, representing an individual's visual interpretation of a concept space or idea. The diagram forms a map of concepts (or nodes) and the relationships between these concepts. Concepts are usually connected in a downward-branching hierarchical structure. Mind maps can be used as an aid in learning and problem solving by facilitating information processing and organisation. Mind maps are also useful for discovering and defining semantic relationships between sub-concepts in a subject area.

Mind mapping in its raw form dates back many hundreds of years. Evidence of such techniques can be seen in Porphyry's Isagoge as he visually represents Aristotle's "Categories". However, mind mapping in its modern form was more recently formalised. Tony Buzan, an English psychologist, researched the use of mind mapping [Buzan 74] [Buzan & Buzan 93] and has developed his own mind mapping software. Mind maps have since been used across academia and industry. Mind mapping systems are considered "Knowledge Representation" tools and can be used to aid the learner in visually representing

and refining their knowledge of a subject area. This also helps to encode the knowledge, promoting its transfer to long term memory. They are also efficient tools for stimulating idea generation.

There are several open-source mind mapping systems currently available which offer a variety of features, implemented in various languages. Three such systems are described below.

Freemind [Freemind] is a mind mapping application integrated with an easy to operate hierarchical editor. Branches of a mind map can be folded and hidden from view for ease of visualisation. Maps can be exported to various formats, including HTML, XHTML, PNG, JPEG, SVG and PDF. Various graphical features are available such as the addition of images and icons to nodes, grouping areas of a mind map in a cloud and graphical links connecting nodes. Hyperlinks to the WWW or local file system can also be added to nodes. Freemind is implemented in Java and available under the GNU GPL [GPL]. This means that the source code is available to use for any desired purpose without commanding a license fee. It also means that any code developed upon, or derived from current Freemind code must also be licensed under GNU-GPL. Freemind is implemented in Java using Swing [Swing], a widget toolkit for the development of graphical user interfaces (GUI). It can be installed on the majority of operating systems including Microsoft Windows, various Linux distributions and Apple OS X.

Labyrinth [Labyrinth] is a lightweight mind mapping tool. The project page states that the system is intended to be as light and intuitive as possible, while still providing a wide range of features. Various graphical features are available such as font styles, size and colour. Images can be attached to nodes and the colour of nodes can be changed. There is even a paintbrush-style drawing tool which allows the creation of images. Maps can be exported as PNG, JPG, SVG or PDF. Labyrinth is implemented in Python and uses the GTK graphics toolkit and Cairo library for it graphics rendering. Labyrinth is intended for use as part of the GNOME [GNOME] desktop project on Linux and Unix operating systems.

The Visual Understanding Environment (VUE) [VUE] project provides flexible tools and processes for integrating digital resources into teaching, learning and research. The open-

source environment combines presentation software with a concept and content mapping application developed at Tufts University. The visual mapping application provides similar functionality to the pure mind mapping tools described above. Graphical features such as the addition of images to nodes and the altering of link and node colours are available. VUE is quite feature rich and provides additional functionality for the creation of ontologies defined using RDF-S or OWL. Maps can also be published to learning environments such as Fedora [Fedora] and Sakai [Sakai]. VUE is implemented in Java and can be installed on the majority of operating systems.

Appendix B – Information Retrieval on the WWW

Introduction

As educators attempt to incorporate TEL offerings into traditional course curricula, the lack of availability of appropriate digital content resources acts as a barrier to adoption [Brusilovsky & Henze 07]. The development of quality TEL resources has proven to be an expensive undertaking [Boyle 03], yet to scalably support next generation functionality such as personalisation and on-demand adaptivity access to large volumes of varied educational content is required. As discussed in chapter 2, the potential value of the vast quantities of content available via the WWW is beginning to be recognised within the TEL movement. Numerous digital content repository initiatives are beginning to accumulate educational resources and web-based publication now produces enormous volumes of content. Much of this content could be used by TEL applications within educational scenarios. However, within the majority of individual digital content repository initiatives there remain insufficient volumes of contributed content to spawn regular and widespread content reuse.

As the web grows it will become ever increasingly difficult for educators to discover and aggregate collections of relevant and useful educational content [Ring & MacLeod 01]. There is, as yet, no centralised method of discovering, aggregating and utilising educational content, from these various disparate sources, within TEL. This research proposes that Information Retrieval (IR) techniques and technologies could be applied to traverse the WWW and centrally collate educational resources, categorised by subject area. These subject specific collections of content could then be used by TEL applications during the generation and execution of learning experiences.

The objective of this chapter is to provide the reader with a detailed insight into the foundations and current trends of Information Retrieval on the WWW. This chapter is used to identify the appropriate techniques and technologies required to support the discovery, classification and harvesting of educational content from open corpus sources. An introduction is provided to the evolution of Information Retrieval and the development of the WWW. The chapter then provides a detailed description of web crawling and web-based IR algorithms which support content discovery on the WWW. The growth of the WWW is then examined. This acts as an introduction to, and foundation for, the analysis of past and current focused crawling techniques and systems which enable the subject specific discovery of content. Content indexing and retrieval are then discussed in relation to web-based content.

Based on the analysis in each section, the chapter concludes by discussing the elements of web-based IR which could be used to influence the discovery and collation of subject-specific educational content for use in TEL.

The Evolution of Information Retrieval

Information retrieval (IR) is a scientific field which evolved in response to the various challenges of locating and accessing relevant, or sought after, information. IR proposed and developed a principled approach to searching for documents, for information within documents and for metadata describing documents. However, IR as a discipline was not conceived during the technological revolution of the late twentieth century. On the contrary, despite the term "Information Retrieval" being first coined by Calvin Mooers in 1950 [Mooers 50] [Mooers 51], methods of defining the interaction between humans and information can be traced back through history to the Library of Alexandria [Alexandria] and beyond, as shown in figure 3-1.

Modern IR is an extremely broad, interdisciplinary field with foundations in Computer Science, Mathematics, Library Science, Information Science, Cognitive Psychology, Linguistics and Statistics among others. The idea of using machines to facilitate the automated search for, and retrieval of, relevant information was first proposed by Vannevar Bush in 1945 [Bush 45]. The first implementations of such mechanised IR systems began to emerge in the 1950s [Kent 64]. Many of the key algorithms and methodologies still in use in the field today were defined more than twenty years ago [Salton 75].

IR systems are generally concerned with receiving a user's information need in textual form and finding relevant documents which satisfy that need from a specific collection of documents. Typically, the information need is expressed as a combination of keywords and a set of constraints. Most IR systems have focused on storing and viewing documents, methods for processing queries and determining document relevance, and user interfaces for querying and refining results [Sampath 1985].

The two metrics generally used in the evaluation of IR systems are *precision* and *recall* [Salton 89]. Precision is the fraction of documents retrieved in response to a query that are relevant to the users information need when making that query. Recall is the fraction of the total set of relevant documents in a collection which are returned for a given query. These

metrics are highly subjective, as relevancy can only be assigned based upon the users intent when submitting a specific query.

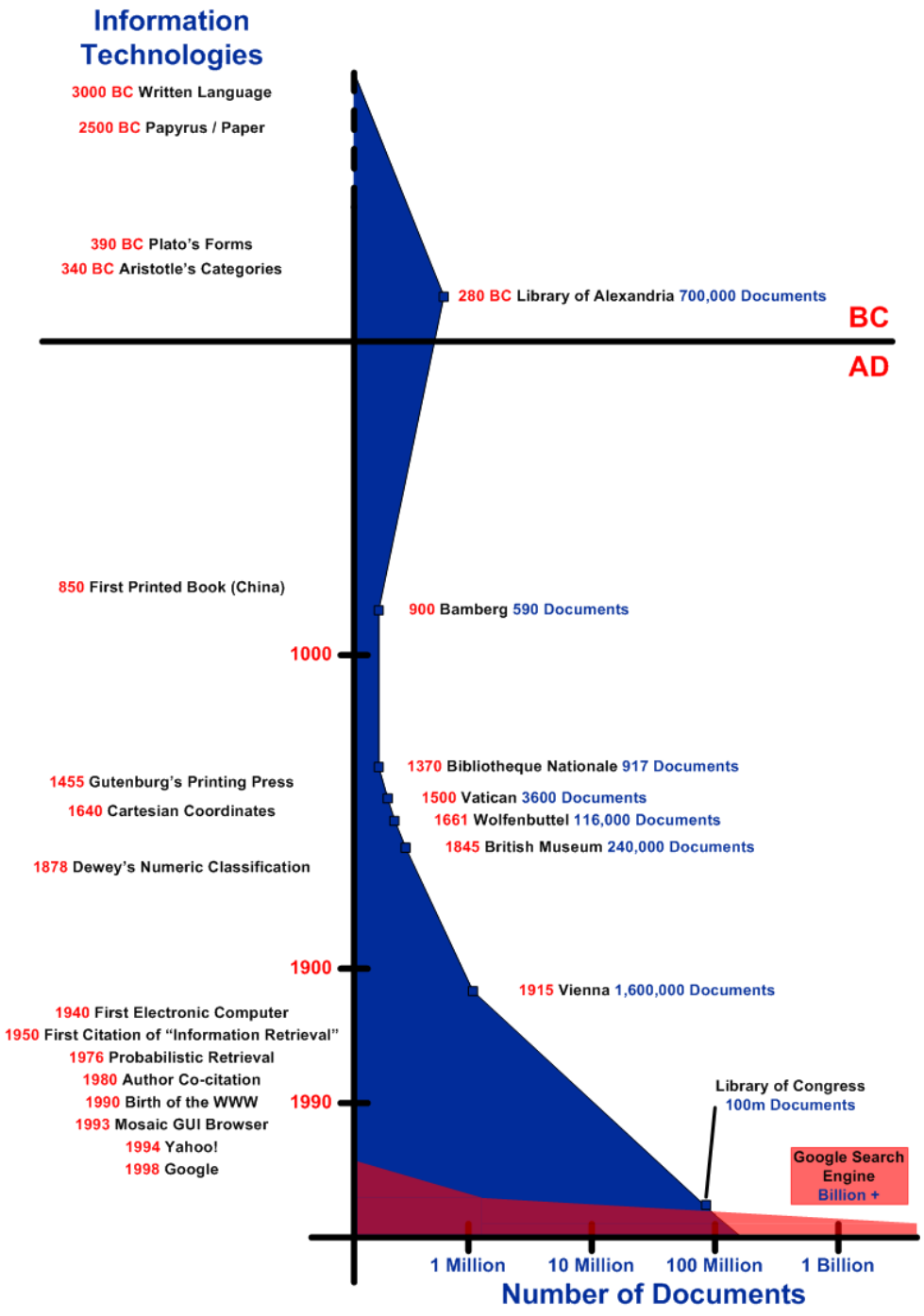


Figure 0-1 Information Retrieval - Timeline10

[10] This diagram is based on a similar image produced in 2002 by Ned Fielden at San Francisco State University. Available online at http://online.sfsu.edu/~foelden/images/timeline.gif

Over the last two decades, with the emergence of the WWW, the field of IR has evolved. What was once a discipline primarily applied in academia, now forms the foundations which underlie most mainstream means of sourcing and accessing information. Much of the research conducted in IR from the 1950s onwards is still applicable to the IR systems in use on the WWW today. However, differences between the structure of the WWW and typical document collections and key differences between the users of web-based IR systems and traditional IR systems challenge many of the assumptions in this early work.

The structure of the document collection alters the metrics which can be used to assess the performance of IR systems. In relation to the WWW, recall is an ineffective means of evaluation, as recall requires the entire collection of relevant documents to be known in advance of each query performed. This calculation cannot be performed on web-based collections as the entire set of relevant pages is unknown. The WWW is constantly growing and evolving, and there is currently no complete index of the entire web in existence.

In 1990 the National Institute of Standards and Technology **(**NIST) [NIST] was asked to build a very large test collection for use in the evaluation of text retrieval technology developed as part of the DARPA TIPSTER project [Harman 92]. This collection was to be of the scale of 1 million full-text documents. This was approximately 100 times larger than any non-proprietary test collection in existence at that time. NIST subsequently proposed that this collection be made available to the full research community by the formation of the Text REtrieval Conference (TREC) [TREC]. For each TREC, NIST provides a test set of documents and information queries. Each participant runs their own retrieval system on this data, and submits a list of the their top-ranked results. NIST pools the individual results, judges the retrieved documents for correctness, and evaluates the results. The first TREC took place in September, 1992 with most leading IR research groups participating. TREC proved a catalyst for research on IR methods that had the ability to scale to potentially enormous corpora such as the WWW. The growing popularity and development of search engines on the WWW only served to increase the demand for scalable IR methodologies even further.

Conventional IR systems traditionally catered for experienced users of the system and domain experts such as librarians and academics. However, as the WWW gained more

popularity and mainstream use, web-based IR systems could not make assumptions regarding the users level of knowledge with regard to the system or the subject domain. The IR system has to cope with users across the spectrum of knowledge and ability [Hölscher & Strube 00].

The World Wide Web and Hypertext

A hypertext, originally conceived by Vannevar Bush [Bush 45] and later coined by Ted Nelson [Nelson 81], can be defined as a collection of information fragments, or nodes, that have active cross-references known as hyperlinks. These links allow an individual browsing a node in the collection to jump to another node in a different location when desired [De Bra et al. 94]. This loosely coupled structural design means that information, or new nodes, can be added to the collection at any point and at any time. However, the majority of early hypertext systems were constructed using individual collections of documents, often in a single subject domain [Coombs 90].

The WWW [Berners-Lee et al. 92] is essentially a distributed hypertext on a massive scale. Nodes can potentially be located anywhere in the world, contributed by millions of authors and consumed by even more readers. However, the drawback to this ease of publication is that there is no organised method to catalogue or list the nodes contained in the collection. The Hypertext Transfer Protocol (HTTP) is used to retrieve individual nodes from the server on which they reside, but there is no protocol for discerning what servers are in existence and what nodes are on each server [De Bra & Post 94a].

By unleashing publication on such an unprecedented scale, the WWW has acted as a principal driver of innovation in the IR field. This explosion of published information would be moot if information, relevant to a user's interests and needs, could not be easily and quickly located. In the early years of the WWW, two resources provided the only real method of finding information; the WWW Virtual Library [VLib] and the NSCA "What's New" page [NSCA]. Both of these resources were essentially catalogues of hyperlinks, in the case of VLib, this catalogue was organised into a hierarchy. However both were manually maintained, and as the growth of the web began to accelerate, it became infeasible to keep these catalogues current and comprehensive [Pinkerton 00].

Web Search Engines were to become the dominant IR mechanism on the WWW. These search engines are services which are largely fed by web-traversing *robots*. By exploiting the

hypertextual nature of the WWW, these *robots* browse nodes on the web, recursively following the hyperlinks contained in each node, archiving content and constructing a database of the pages encountered. These *robots* later become known by various pseudonyms such as Web Spiders, Web Crawlers, Web Robots and Web Scanners[11].

Traditional search engine architectures have three main components: the web crawler, which traverses the web in an attempt to discover new or modified content; the indexer which creates a searchable reference of all the content encountered; and the user interface which allows the individual to express their information need and filter through the content returned. To fully understand how these IR services have evolved and the means by which they function, it is necessary to examine their emergence and development over the past 15 years.

Web Crawling and Web Search

Solutions to the problem of finding information on the rapidly expanding, yet still very young, WWW began to arrive as early as 1993. The WWW Wanderer was developed in June 1993 by Matthew Gray at MIT and became the first means of searching for information on the web without referring to a manually maintained index. Gray initially set out to measure the growth of the WWW and created the wanderer to count active web servers [Gray 95]. However, the wanderer was soon upgraded to capture and store the URL's it encountered. The WWW Wanderer employed a web crawler to retrieve the URLs of hyperlinks on the WWW, and used these links to create a central index, a truly pioneering technique which is still in mainstream use today. This database of content links generated by the WWW Wanderer became known as the Wandex and is widely regarded as the first search engine.

Three further searchable indexes of the web, or parts of the web, based upon web crawling technology had emerged by 1994: Jumpstation, developed by Jonathan Fletcher at the University of Sterling in Scotland; the Repository-Based Software Engineering (RBSE) index [Eichmann et al. 94]; and the WWW Worm [McBryan 94]. Each of these systems employed a crawler to retrieve the content of pages on the web, and used this content to create a central index.

[11] In the context of this Thesis such archival *robots* will, from this point onwards, be referred to as crawlers.

The IR services based upon these early web crawlers struggled, and eventually failed, for various reasons. Most importantly the infrastructure of the WWW was at an extremely early stage of development, and as a result, bandwidth, processing power and storage were all still at a premium. Many website owners felt that the WWW Wanderer was consuming too much bandwidth and adversely affecting the performance of their sites. Gray re-wrote the crawler to follow a breadth-first algorithm which is more efficient on the WWW and allows a better spread of resource demands across web servers. This demand for bandwidth and storage made it infeasible for a crawler to download the entire content of each node encountered on the WWW. In an effort to circumvent these issues, Jumpstation and the WWW Worm downloaded and indexed only the title of pages encountered on a crawl. They also listed the output of any searches performed over the index in discovery order rather than implementing any ranking mechanism. RBSE adopted an alternate approach by restricting crawls to a small portion of the WWW. Neither approach provided a comprehensive service for the discovery of content across the entire WWW.

WebCrawler was developed by Brian Pinkerton at the University of Washington in 1994 [Pinkerton 00]. WebCrawler was the first crawler to create an index using the entire content of each page encountered rather than just the URL or the hyperlink anchor text. WebCrawler became massively popular and had served 1 million searches within seven months of its launch[12]. WebCrawler also foreshadowed the importance that hyperlinks themselves were to play in the emergence of search algorithms. Pinkerton ran tests upon the WebCrawler index to determine which sites had the most incoming hyperlinks from other sites, not a significant leap from the Pagerank and Hubs and Authorities algorithms which will be discussed in section 3.4.1 below. The site with the most incoming links in April 1994 was the website of the World Wide Web Project at CERN [Battelle 05].

The first truly web-scale search engine was Alta Vista, developed by Louis Monier for Digital Equipment Corporation (DEC) in 1994 [AltaVista]. DEC was largely a hardware company, and it had just launched a new line of high-powered processors. Alta Vista was used as a test for the new machines and it was this processing power that set it apart from the growing crowd. Web crawlers work in a linear fashion, discovering and following hyperlinks

[12] The one millionth search performed was, somewhat worryingly, for "Nuclear Weapons Design and Research" [WebCrawler].

in a recursive fashion. This is a restrictive process if the aim of the crawler is to traverse the entire WWW. Alta Vista surmounted this challenge by running upwards of 1000 crawlers in tandem, each on different sections of the web. This allowed it to generate the closest thing to a complete index of the WWW yet created; 16 million documents at its December 1995 public launch date [Battelle 05]. Within a year Alta Vista had served more than 4 billion queries.

Lycos was also created in 1994 [Lycos], but at Carnegie Mellon University (CMU) by Dr. Michael Mauldin. Lycos used a web crawler to traverse the web in a similar fashion to the systems mentioned previously. However it had one innovative difference which was to influence all future successful crawlers. Lycos was the first search engine to use the hyperlinks into a webpage to determine its relevance to a query. The anchor text of each link to a page was analysed and used in an attempt to deduce the context and meaning of the page. Lycos was also the first search engine to display summaries under each result in its list, rather than just a list of URLs.

Web crawlers have since evolved to feed a host of web IR tools. The two most popular web-based IR systems are Google [Google] and Yahoo! [Yahoo!]. Yahoo! was created, also in 1994, by Jerry Yang and David Filo, PhD candidates at Stanford University in California. The focus of Yahoo! was subtly different to the majority of other search engines. Yahoo! is essentially an automated directory of the web. Web crawlers are used to traverse the web discovering content which is then sorted into hierarchical categories based on subject.

Google was founded in 1998 by Larry Page and Sergey Brin, also PhD candidates at Stanford University in California. Google is a search engine in the more traditional sense of the name. The site employs an army of crawlers to archive and index portions of the WWW. These indexes are then combined and deployed via a web interface which allows an individual to search for content across the entire index using certain terms or phrases contained within the desired information. Google enjoys a dominant share of web-based IR traffic, and has done for a number of years. The market share statistics from comScore [comScore] for June 2008 show Google with 61.5% of web search traffic.

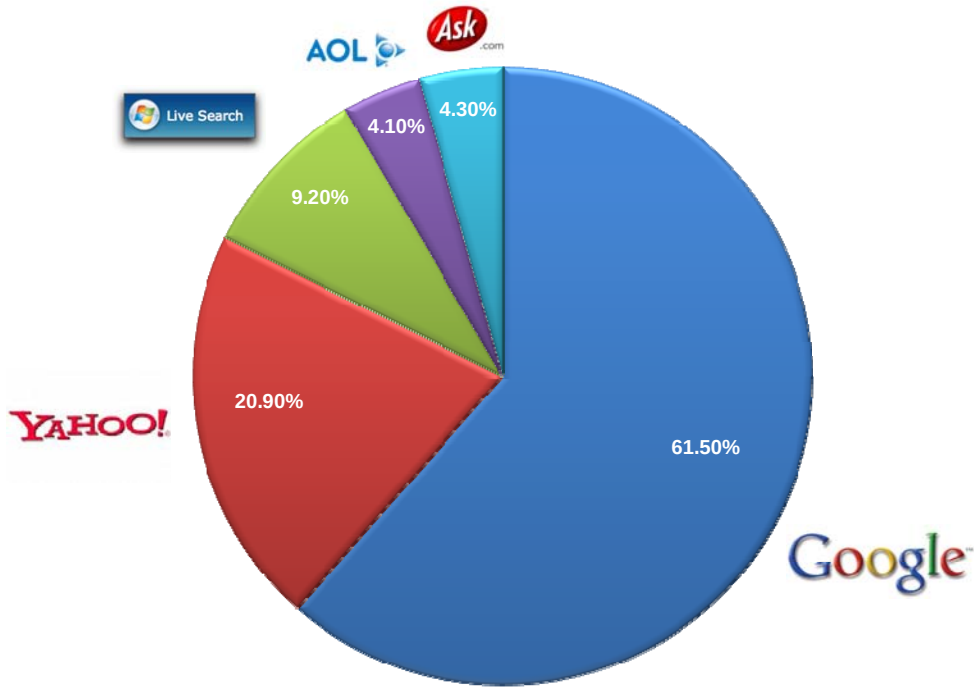


Figure 0-2 Search Market Share – June 2008

It was the emergence of two algorithms which exploit the hyperlinked structure of the web which allowed Google to gain such dominance over its rivals, despite its late arrival upon the WWW IR scene. An examination of these algorithms will give some insight into the structure of the WWW and how this structure can be exploited when conducting focused crawls for particular subject areas and domains, as will be discussed in section 3.6.

Web-based IR Algorithms

The practice of link analysis on the WWW has its foundations in the area of Bibliometrics. Bibliometrics was originally defined as "the application of mathematics and statistical methods to books and other communication media" [Pritchard 69]. The research field has since been further refined to encompass "the mathematical study of libraries and bibliographies" [Egghe & Rousseau 90].

Citation analysis is one of the most commonly used Bibliometric methods. It involves the study and analysis of citations in published literature to produce quantitative estimates of the importance and impact of individual scientific papers and journals [Kleinberg 99b]. Blaise Cronin referred to citations as "frozen footprints on the landscape of scholarly achievement"

[Cronin 84]. The most well-known measure in this field is Garfield's impact factor [Garfield 72], used to provide a numerical assessment of journals. The impact factor of a selected journal is calculated as the average number of citations received by papers published in that journal over the previous two years [Egghe & Rousseau 90].

Pinski and Narin [Pinski & Narin 76] proposed a more subtle citation-based measure of publication impact. This improved measure is based upon the observation that not all citations are of equal importance. Pinski and Narin argue that a journal can be deemed influential if it is heavily cited by other influential journals. Put simply, they state that a citation from an influential, high profile journal should carry more weight in the generation of a journals impact factor.

Hubs and Authorities

Jon Kleinberg noticed the parallels between the hyperlinked structure of the WWW and the bibliographical structure of academic publications [Kleinberg 99b] when conducting research into CLEVER, a search engine that aimed to exploit the hyperlinked nature of the WWW [CLEVER], at IBM's Almaden Research Centre. Kleinberg defines the WWW as "an intricate form of populist hypermedia, in which millions of participants, with diverse and often conflicting goals, are continuously creating hyperlinked content". The application of bibliometric methods to the study of the WWW is termed webometrics. The formal definition of webometrics is "the study of the quantitative aspects of the construction and use of information resources, structures and technologies on the WWW drawing on bibliometric and informetric approaches" [Almind & Ingwersen 97] [Björneborn & Ingwersen 04].

The analogies between the citation structure of scholarly journals and the WWW are immediately obvious. The creation of a hyperlink between two pages, *A* and *B*, on the WWW can be used to infer the following: The author of page *A*, by creating a link to page *B*, has conferred some measure of confidence and authority on *B*. The author of *A* is stating that they believe *B* to be a relevant and reputable resource in relation to the subject under examination.

Pinski and Narin's extension to citation analysis can also be applied to hyperlink analysis on the WWW. A hyperlink to a page is more important if it comes from a highly respected, influential web-page. Links to a page *A* from the BBC, CNN and Yahoo! homepages potentially make *A* more important that a page *B* which is linked to from 100 small, obscure

websites. This pattern holds true, particularly in non-commercial environments, such as educational or informational web pages where high quality resources tend to link to other authoritative sources of information on that subject. The pattern differs in commercial environments where it is common for authorities not to link to the other main authorities in their area, as they are often in direct competition with these websites. This theory of web-graph structure is known as "Hubs and Authorities" [Kleinberg 99a].

Kleinberg identifies a diversity of roles among web pages in a common subject area. Pages which have a large volume of incoming links, particularly from other influential web pages, prove to be the most prominent sources of information on a particular subject. Pages which fall into this category are labelled "authorities". Other pages, which are equally intrinsic to the structure of the WWW, offer collections of links to numerous recommended, high quality sites, or authorities, in a particular subject area. These pages act as reputable resource lists and are labelled "hubs". The nature of the relationship between hubs and authorities is very asymmetrical. Hubs link heavily to authorities but may themselves have very few incoming links.

If page A links to page B, it can be inferred that A is recommending B. If A is linked to by a large number of influential documents in a subject domain, then A is an authority on that subject. If B links to a large number of influential documents in a subject domain, then B is a hub for that subject domain. A good authority will be linked to by many good hubs, and a good subject hub will link to many good authorities.

Hyperlink-Induced Topic Search (HITS) is a ranking algorithm developed using the hubs and authorities theory. In response to an IR query a text based search is invoked. HITS is then applied to a small subgraph of the web constructed from the search results. A hub weight and authority weight are calculated for each node in the subgraph. A node's authority weight is proportional to the sum of the hub weights of nodes *which link to it*. A node's hub weight is proportional to the sum of the authority weights of nodes *which it links to*.

PageRank

PageRank is another algorithm based upon these structural patterns. It was developed by Larry Page and Sergey Brin at Stanford University between 1995 and 1998 and went on to form the basis of the Google [Google] search engine. PageRank was developed as an attempt

to calculate a global ranking of every page on the WWW, regardless of its content, based solely on the page's location in the Web's graph structure [Page et al. 98]. The algorithm extends the notion of "authorities" in Kleinberg's work.

PageRank interprets a hyperlink from page *A* to page *B* as a vote by page *A* for page *B*. The algorithm also analyses the page from which the hyperlink has emanated. Links from pages that are deemed important in their own right are more heavily weighted and have more influence on the resulting rank calculated for the linked to page.

PageRank is a probability distribution and is used to represent the likelihood that a random surfer on the WWW will arrive at a particular page. The numerical rating can be calculated for any collection of pages regardless of size, which allows it to be deployed against the open web. The calculation of a rank requires several iterations through the collection to ensure the assigned values are as close to true as possible. A probability is expressed as a numeric value between 0 and 1. A value of 0.5 assigned to an event is commonly referred to as a "50% chance" of that event taking place. Therefore a PageRank of 0.5 assigned to a page implies there is a 50% chance that a web surfer randomly clicking on links will end up on this page.

Theoretically, the sum of all the PageRank scores in a collection should add up to one. In the first iteration, the probability is evenly divided between all the pages in the collection. Hence, in a ten page collection, each page would begin with an estimated PageRank of 0.1. Each link between pages confers some value. The PageRank conferred by an outbound link **L** is equal to the document's own PageRank score divided by its normalised number of outbound links. It should be noted that links to specific URLs count only once per page. This means that the PageRank value for any page p_i can be expressed as the PageRank values for each page p_x out of the set $M(p_i)$, which contains all pages that link to page p_i, divided by the number of links emanating from page p_x.

$$PR(p_i) = \sum_{p_x \in M(p_i)} \frac{PR(p_x)}{L(p_x)}$$

However, the PageRank theory states that even an imaginary web surfer randomly clicking on links will stop clicking at some point. The probability, at any point of the calculation, that

the surfer will continue clicking is called the damping factor, or *d*. The damping factor is subtracted from 1 and the result is divided by the number of documents in the collection.

If a page has no links to other pages, it acts as a "cul de sac" and terminates the random surfing process. To avoid this trap, pages with no outbound links are assumed to link to all other pages in the collection. Their PageRank scores are thus evenly divided among all other pages.

$$PR(p_i) = \frac{1-d}{N} + d \sum_{p_x \in M(p_i)} \frac{PR(p_x)}{L(p_x)}$$

Analysis

The main disadvantage of link analysis algorithms such as HITS and PageRank is that they favour older pages. New pages, even if their content is of very high quality, will initially not have many incoming links unless the page is part of an existing site's link structure.

Various methods of manipulating link analysis algorithms have been attempted in an effort by commercial sites to improve search results rankings and monetise advertising links. These strategies have impacted the effectiveness of such algorithms when used in isolation. Most search engines use a combination of factors such as traffic volumes and click through rates in combination with these algorithms. Search engines are also known to actively penalise sites designed to artificially inflate their ranking. In December 2007 Google began actively penalising sites selling paid text links [Link Buying].

Although link analysis algorithms are primarily used as a ranking methodology for the serving of search results, they can also have implications upon the way the web is crawled for information. The structure of the web described by these algorithms can be exploited when crawling the web for content in a particular subject area. This directly influences the design of an application which supports the discovery, classification, harvesting and delivery of topic-specific educational content from open corpus sources, which is a goal of this research. The exploitation of link structure in web crawling will be discussed in more detail in section 3.6.

Summary

The hypertext structure of the WWW reflects bibliometric patterns, and can be compared to that of a scholarly journal. Quality resources tend to link to other authoritative sources in the same subject area, much like a reference in a journal paper. This pattern is detailed in the Hubs and Authorities theory and exploited by algorithms such as HITS and PageRank. Webometrics is a field of study which examines the construction and use of content and technologies on the WWW drawing on bibliometric and informetric approaches. Exploiting this structural pattern and implementing IR techniques based on bibliometrics could be extremely useful for discovering quality sources of educational content.

The previous two sections described how web crawling and web search methodologies were developed during the emergence of the WWW and how these solutions have evolved over the course of time. The following section will introduce recent trends in the growth of the WWW, and how this growth has affected the ability of IR tools to perform efficiently and effectively. This section will also explain how this growth has promoted the development of a new form of web crawling and IR on the web called "focused crawling".

WWW Growth and the Deep Web

In 1999, Inktomi [Inktomi] was utilising a cluster of several hundred individual servers, each with 75 GB of RAM and over 1TB of spinning disk, phenomenal processing power for the time. Using these resources, the crawler was able to process over 10 Million web pages each day. However, even in 1999, this accounted for a mere 30-40% of the known web, the size of which has grown exponentially since [Chakrabarti et al. 99]. The number of servers modern web-scale search engines have to deploy to handle the sheer volume of data available on the WWW is unknown, but many estimates state that Google have upwards of 1,000,000 web servers in operation [ZDNet].

The Deep Web

As publishing on the WWW becomes ever easier, as discussed in section 2.2.5, and the growth of the web accelerates, tracking the emergence of new information, and sourcing specific information within this distributed network of nodes and servers becomes increasingly difficult. The loosely coupled design of the WWW also means that some nodes on the network may have no incoming links and thus are essentially "undiscoverable" by conventional web crawlers. This portion of the web, unreachable when employing link

analysis algorithms, has become known as the "Deep Web". In 2001 it was estimated that public information on the deep web was between 400 and 550 times larger than the mainstream WWW [Bergman 01].

One of the reasons a page may have no incoming links is if the content is dynamically generated. An example of this is commercial web pages which allow the user to submit a query in order to retrieve a list of products. The required page is then dynamically constructed, usually by querying an internal database of potential information. Various web crawling techniques have been developed in an attempt to access such dynamic content. One approach is to generate and submit sets of keywords to site query boxes to generate and retrieve most of the potential dynamically generated pages [Raghavan & Garcia-Molina 01] [De Carvalho Fontes & Silva 04] [Ntoulas et al. 05]. A second approach is to simulate an individual's browsing behaviour on the WWW using agents [Lage et al. 04].

In a study on the hypertext connectivity of the WWW [Broder et al. 00], web pages were categorised based upon their link structure. Three main categories and two sub-categories were defined. The analysis was conducted using over 200 million individual pages and approximately 1.5 billion hyperlinks collected by a web crawl in 1999. This approach to WWW link analysis has become known as "BowTie Graphy Theory", for reasons which become obvious upon viewing the graphical representation produced by the study.

The Strongly Connected Component (*SCC*) category consists of pages that can reach one another along directed links. The second and third categories are called *IN* and *OUT*. IN consists of any pages that can reach SCC pages but cannot be reached by any SCC pages. Out consists of pages that can be reached by SCC pages, but do not link back to any SCC pages respectively. The rest of the pages, called Disconnected Components, cannot be reached and do not reach any SCC pages. In theory an individual or web crawler browsing the WWW can pass from any page within the IN category, through SCC to any page in OUT. Hanging off IN and OUT are Tendrils. These Tendrils contain pages that are reachable from other pages within IN, or that can reach portions of OUT, but which do not link to SCC. It is possible for a Tendril page from IN to link to a Tendril page which leads into OUT, forming a Tube.

Of the 200 million web pages analysed [Broder et al. 00] over 56 million, or 27%, were categorised as strongly connected and were placed in the SCC. 43 million, or 21%, were

found to link into the SCC set but were not linked to by the SCC set and such were placed in IN. Similarly, 43 million, or 21%, were linked to by pages in the SCC set but did not link back to any pages in the SCC set and were placed in OUT. Another 43 million were found to link to other pages within their category but never to the SCC and were placed in Tendrils. Finally 16 million pages were found to be Disconnected Components. This graphical description of the WWW explains why it can be difficult for web crawlers to discover and access significant portions of the web. If the crawler starts its navigation from a page in the OUT set, it can only reach approximately one fifth of whole WWW.

All the information that a search engine is unable to access has been termed "Dark Matter" [Bailey et al. 00]. This not only includes the deep web, but also pages which are intentionally hidden from web crawlers through the Robots Exclusion Protocol [Robots.txt]. The Robot Exclusion Protocol [Robots.txt], also known as the Robots Exclusion Standard or robots.txt, was established as an independent method of defining web crawler "politeness" by the members of the robots mailing list, robots-request@nexor.co.uk, in June 1994. The protocol is a convention by which cooperative, or polite, web crawlers can be requested not to access all, or parts, of a website which is otherwise publicly accessible.

A file, called robots.txt, can be specified for each domain name on the WWW to provide instructions, regarding the content and structure of the site, to web crawlers. This file should be downloaded and analysed by a crawler before any download requests are made to the site. However, it is at the discretion of each crawler as to whether they obey the instructions contained within robots.txt files. Crawlers which disregard these politeness requests are often targeted by spider traps or have their IP addresses blocked entirely by web servers.

WWW Growth

The growth of the WWW has been rapidly increasing, particularly since the turn of the century, and the exact size of the web has become a contentious issue. Two of the main publishers of statistical information regarding the WWW are Netcraft [Netcraft] and the Internet Systems Consortium (ISC) [ISC]. Some estimates place the creation of new pages on the WWW at 320 million per week [Ntoulas et al. 04].

Netcraft publish statistics on the number of web servers detected each month. While Netcraft do not publically state how they generate their figures, it is widely believed that they use a

web crawler to traverse the web and send requests to each server encountered. In the Netcraft statistics for April 2008, they estimate that there are a total of 165,719,150 web servers on the WWW, of which approx 70 million are active.

The ISC produce statistics which measure the number of individual hosts on the WWW. This is accomplished through a domain name survey, calculating the number of IP addresses that have been assigned. The ISC statistics for January 2008 reported 541,677,360 hosts on the WWW. The essential difference between the Netcraft and ISC metrics is that the ISC include both producers and consumers of information in their calculations, whereas Netcraft include only producers of information.

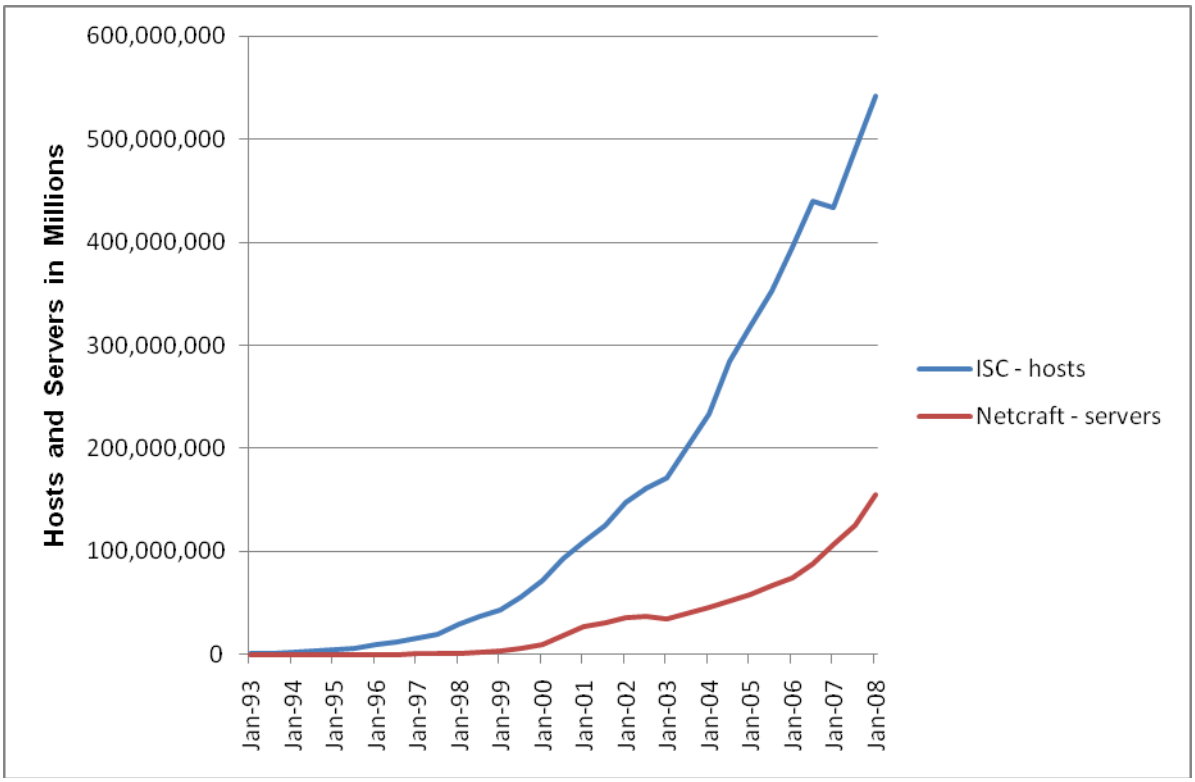


Figure 0-3 WWW Growth Statistics 1993-2008

While the size of the WWW has been rapidly increasing, similarly has the number of people using this vast network. Internet World Stats [IWS] publish statistics on the number of people worldwide accessing the WWW. This has grown from an approximate figure of 16 million in December 1995 to over 1.3 billion today.

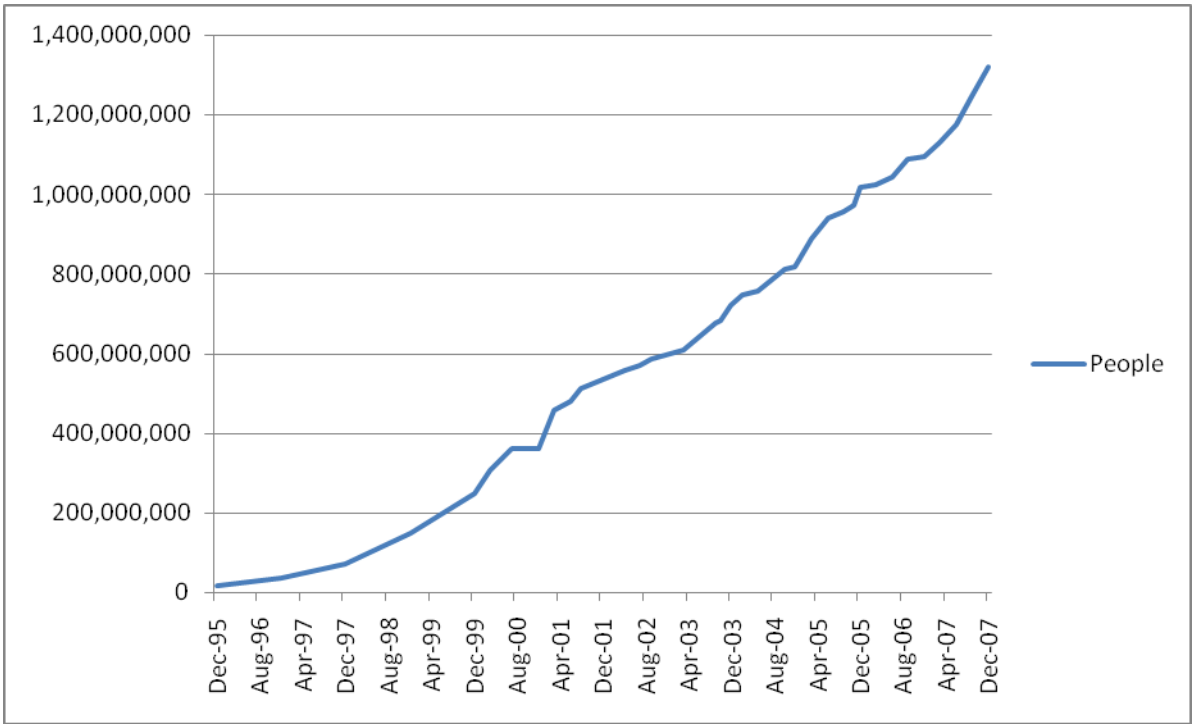


Figure 0-4 WWW User Statistics 1995-2007

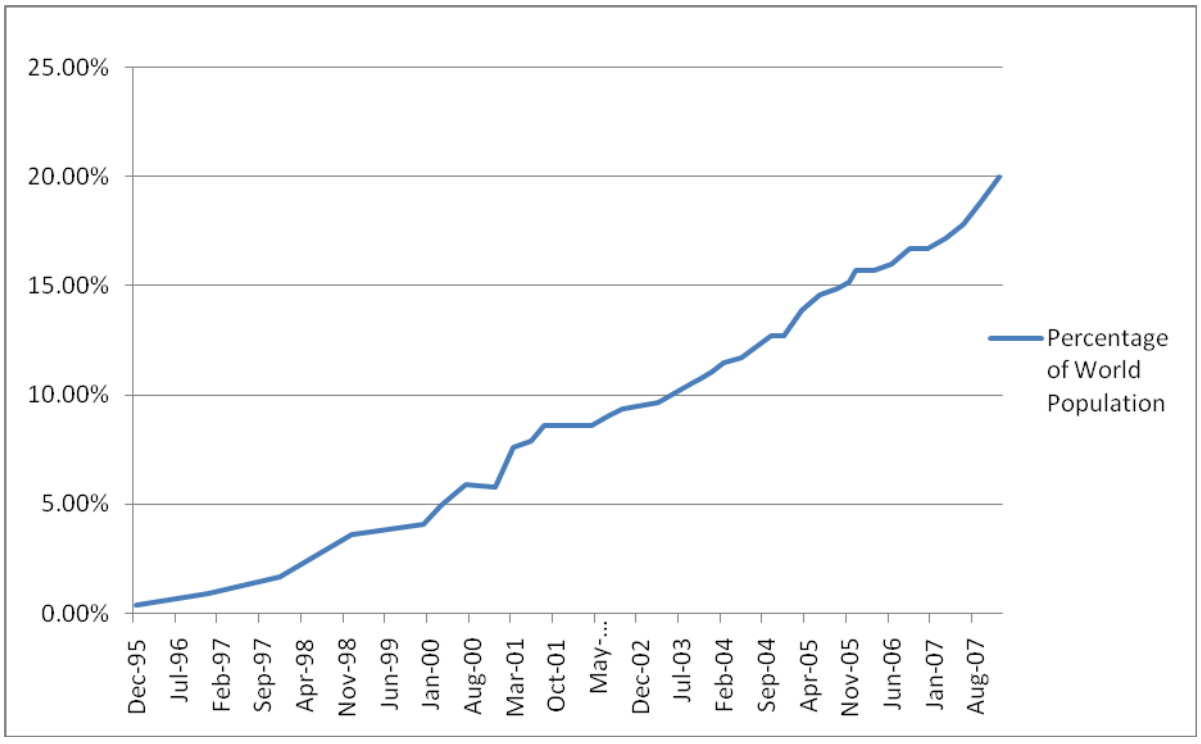


Figure 0-5 WWW Users as a Percentage of World Population 1995-2007

This rapid growth of the WWW has been exacerbated by the recent explosion of web content publication beyond the traditional territories of North America and Europe, and beyond the traditional social strata of academia and industry. The WWW user base conventionally had a

defined split between producers and consumers of information. However, the ease with which information can now be published on the WWW and the relative prevalence of broadband access has resulted in the emergence of a new type of user, the pro-sumer. These users both produce and consume content. The steady growth of this type of user is directly linked to the proliferation of media sharing sites, blogs, social networking and in the field of education, to digital content repositories.

Analysis

The overwhelming challenge of attempting to source specific content on the ever expanding WWW is, in part, due to a traditional one-size-fits-all philosophy. Google and many similar search engines which utilise web crawlers attempt to cater for every possible search that may be performed. Although such services are invaluable due to the broad spectrum of information they cover, the resulting diversity of content often results in thousands of responses of little relevance or quality to all but the most expertly constructed queries.

As a result of this volume and diversity of information, the individual can suffer from information overload as they attempt to comprehend the expanse of unstructured material being presented. They may become disoriented, unable to determine how the current information displayed relates to the information they seek [Eklund 95]. This phenomenon has become known as 'Lost in Hyperspace' [Conklin 87] [Edwards & Hardman 89], and is a frequent occurrence in situations where people are exposed to large volumes of apparently disorganised information. Even experienced users can become spatially confused when presented with information whose structure is seemingly random and incoherent [Laurillard 93] [Theng 97].

This becomes an even more acute problem when examined within the scope of education. In learner-driven educational experiences, it can often be extremely difficult for the learner to filter the information which they encounter. This is the case not only with regard to the relevancy and applicability of new information, but also with regard to the quality of the on-topic information. It is not sufficient in an educational experience to provide a student with a mixture of on and off topic content, of high and poor quality content. Students can be distracted and become disengaged from the educational experience by encountering irrelevant information or due to frustration with the quality of content encountered.

More notice is now being given to the fact that the massive scale of traditional web IR solutions can also be their biggest disadvantage. There is a growing sense of natural limits within the web IR industry, a recognition that in certain circumstances covering a single galaxy can be more practical, and useful, than trying to cover the entire universe [Gillmor 98]. This realisation has resulted in the emergence of a new approach to web crawling, termed "focused crawling". This is a methodology whereby crawls are conducted for specific scenarios or subject areas using automatic filtering of the content encountered to estimate relevance. This can be used to generate smaller, more specific content collections which can be accessed through topical portals.

It is the aim of this research to collate and deliver quality, topic-specific collections of educational content to both the educator and learner within a technology enhanced learning environment. This content should be filtered to ensure that only content that is relevant to a specific subject domain or scenario is included in the pool. Focused crawling can provide both a means of content discovery and filtration for the creation of these content collections. The state of the art of focused crawling will be discussed in more detail in the following section.

Summary

This section discussed the recent surge in the growth of the WWW. The sheer volume of information available via the WWW makes it increasingly difficult to source relevant information. This has negatively affected the ability of web-based IR tools to perform efficiently and effectively. This section also introduced the concept of the deep web and the loosely coupled connectivity of the WWW. The following section aims to provide the reader with a detailed introduction to focused crawling and the various techniques of employed to efficiently perform topic-specific content discovery.

Focused Crawling

The goal of a focused web crawler is to selectively seek out pages that are relevant to a pre-defined set of topics. A focused crawler can be defined as a web crawler which actively seeks, acquires, indexes and maintains pages on a specific topic which represent a relatively narrow segment of the WWW [Chakrabarti et al. 99].

Besides sourcing content based on its content, focused crawling provides another benefit by allowing a web crawler to process specific sites to greater depths than general purpose crawlers. Focused crawlers can spend more time perusing highly relevant sites rather than attempting to attain broad coverage of the entire WWW in a breadth-first manner. As a result, highly relevant pages can be discovered that may have been overlooked by more general-purpose crawlers [Chakrabarti et al. 99]. In highly specific subject domains, link popularity, which helps to guide the majority of general purpose search engines, is not always strongly correlated to content relevance and quality [Marchiori 97]. This meant that focused crawlers required new methods of analysing the WWW to discover links to relevant content. The various methods employed to achieve this by focused crawling systems are discussed in detail below. However, before discussing individual methods, Topical Locality, a theory which underlies almost all focused crawling techniques will be explained.

The various methods employed by focused crawling systems to facilitate the discovery of topic-specific content on the WWW are discussed in detail in the sections below. However, before discussing individual approaches, Topical Locality, a theory which underlies almost all focused crawling techniques will be explained.

Topical Locality

Topical locality refers to the observation that web pages are typically linked to other pages with semantically similar content. In other words, web pages tend to link to other pages which contain related information. Focused crawlers exploit this phenomenon to aid the topic-oriented discovery of web pages.

In an examination of this theory [Davison 00], topical locality was combined with an analysis of the anchor text of hyperlinks as a discriminator of the relevance of unseen pages. Evaluations were conducted on a random selection of 100000 pages from the archive of the DiscoWeb search engine. The evaluation results showed that a page is significantly more likely to be topically related to the pages to which it is linked, as opposed to other nearby pages or randomly selected pages. Sibling pages are also more topically similar when the links on the parent are physically located close together. The results also found that anchor text is often very informative about the contents of the page it references, and as a result can be useful in discriminating among unseen child pages. Davison found that the inclusion of text in close proximity to the hyperlink did not significantly improve similarity measures.

Similar conclusions were drawn in an evaluation conducted in [Menczer 04]. Topical locality was formalised as two conjectures. The "link-content conjecture" states that a page is likely to be similar in subject matter to the pages that link to it. The "link-cluster conjecture" states that pages about the same topic are likely to be clustered together [Menczer 01], i.e. the content of a page can be inferred not only by examining the pages that link to it, but also by examining its neighbours. An experiment was conducted to validate these formalisations by analysing the correlation between lexical similarity and link distance. The evaluation showed that the lexical similarity of two pages exponentially decays as the range of links between the pages increases. The probability of a page being topically relevant is high within a radius of three links. It then decays rapidly. This indicates that the performance of a focused web crawler can be significantly affected by the proximity of clusters of relevant pages.

Methods of Focused Crawling

The first topic-specific or focused crawling technique dates back as far as 1994, when many of the early general purpose web crawlers were still emerging. However, it was to be upwards of five years before the next developments in the area appeared. Since then, many approaches to focused crawling have emerged, all of which use different techniques to aid the crawler as it selects paths through the WWW to relevant content. A number of current approaches to focused crawling are discussed below.

Link Prioritisation

The link prioritisation approach to focused crawling attempts to order the URLs to download so that the most desirable or relevant pages are downloaded first. Fish-Search [De Bra & Post 94b] and Shark-Search [Hersovicia et al. 98] were two of the earliest systems to attempt to prioritise the URL queue for a focused web crawl to improve the efficiency of the crawl.

The Fish-Search algorithm [De Bra & Post 94a] conducts a type of focused web crawl for each search query. It takes one or more starting URLs and the user's query as input. The queue of URLs to be downloaded takes the form of a prioritised list. The first URL in the list is taken and downloaded in a recursive fashion. The text of each page is analysed and assigned a rating in relation to the user's query. This rating dictates whether a page is deemed relevant and whether its links are scheduled for download. This was an extremely innovative technique and was much more efficient method of crawling for a specific topic than a

traditional breadth-first crawl. However, as a crawl was conducted for each search performed, it placed a very heavy load on web servers [Micarelli & Gasparetti 07].

Shark-Search [Hersovici et al. 98] is a more aggressive variant of the Fish-Search algorithm. In Fish-Search, regions of the WWW where relevant pages are not quickly discovered are discontinued. Shark-Search overcomes some of the limitations of this approach by measuring page relevance more precisely than the binary relevance function in Fish-Search. Shark-Search also makes finer estimates of the relevance of neighbouring pages and prioritises relevant pages or pages that are most likely to lead to relevant pages. In Shark-Search the potential relevance score of a link is influenced by its anchor text, the text surrounding the hyperlinks. Relevance is also influence by an inherited score from incoming links.

WTMS [Mukherjea 00] is a system for gathering and analysing collections of web pages on related topics. WTMS contains a focused crawling component. This focused crawler assigns each page a representative document vector (RDV) based of the frequently occurring keywords in their URLs. A vector space model is used to compare discovered pages to a set of user-defined seed pages. URLs from pages which attain a similarity score above a specific threshold are queued for download. WTMS exploits the theory of topic locality through the inclusion of a “nearness” coefficient.

WTMS attempts to strike a balance in its estimation of nearness. If strict criteria for determining the nearness between two pages are used, the number of pages downloaded will be lower and some relevant pages may be missed. Conversely, lenient criteria will result in more pages retrieved but at the cost of increasing the number of downloads and the possible dilution of relevant content. The criteria used by WTMS are those shown in the above graph, figure 3-8. Only pages within the parent and sibling sub-trees of a site graph are downloaded as these are considered near. In the example displayed, when the focused crawler resides at page B, then only pages within the directories A, C, D, E, F and G are considered near and scheduled for download. This was believed to be the optimum setting to capture most of the relevant documents within a site without having to download every page on the site.

Other approaches to prioritising the crawl order of unvisited URLs have been researched and comparatively evaluated [Cho et al. 98]. Breadth-first is a common algorithm used in the IR community. If the WWW is represented as a graph of nodes, symbolising pages, and edges,

symbolising hyperlinks then the order in which pages are visited is dictated by page depth. Pages at the same depth as the current page are visited before delving any deeper into a website. The Backlink count metric ranks each page in importance based upon the number of other crawled pages which link to it. This is a simple web-based implementation of bibliometrics. PageRank is a more sophisticated bibliometric method which uses an iterative algorithm to assign importance to each page based not only on the number of backlinks it receives, but also taking into account the relative importance of each page which links to it. PageRank has been examined in more detail in section 3.4.1.

Comparative evaluation of these approaches demonstrates that PageRank outperforms both Backlink and the random ordering if the goal is to crawl the most popular pages. Backlink tends to be more influenced by the seed set of URLs, becoming very locally focused and not discovering sites outside of the clusters where the seed URLs are located. However, if the goal is the discovery of pages relevant to a specific topic, the evaluation shows that Breadth-first performs most successfully [Cho et al. 98] [Micarelli & Gasparetti 07].

Taxonomic Crawling

A much more comprehensive approach to focused crawling on the WWW was proposed in [Chakrabarti et al. 99]. This focused crawling system has three constituent elements: a classifier, a distiller and a crawler. The classifier assesses the text of all the pages discovered for relevance to the purpose of the crawl. The distiller attempts to identify hubs through an implementation of the HITS algorithm [Kleinberg 99b]. Hubs are pages which provide links to many authoritative sources on the topic in question. The crawler conducts the content download, link extraction and manages URL prioritisation. It is governed by both the classifier and the distiller.

A canonical classification tree is generated from a topic taxonomy. The user can select the most applicable nodes, or leaves, from this tree in relation to the subject of the crawl. Once content is discovered and analysed, the classifier places it within the best matching leaf of the tree. If this leaf has been tagged by the user then the content is deemed relevant and its links are scheduled for download in the URI queue.

This focused crawling approach has since been further refined with the addition of a fourth component, the apprentice [Chakrabarti et al. 02]. The apprentice acts as a second classifier

which prioritises the URLs in the queue, or crawl frontier. The idea behind the apprentice is to try and emulate human behaviour when browsing the web. On the WWW "every click on a link is a leap of faith" [Leiberman et al. 01], however humans are much more adept at using a variety of visual and textual clues to estimate the worth of the target page. The apprentice extracts features relating to a link from the Document Object Model (DOM) of the page from which the link was sourced. It then takes these features into account when prioritising the URLs in the crawl queue.

Context Graphs

Contextual graphs provide an alternate approach to the prioritisation of pages along a crawl path [Diligenti et al. 00]. An algorithm is used to build a representative model of the pages that occur within a defined link distance of a set of target pages on a specific topic. Using existing search engines, the WWW is *backcrawled* to build the context model. In other words, searches are performed to find pages which link to the seed set of target pages. A context graph is created which consists of all the discovered pages and their links in relation to the target pages. A relational value is calculated for each page. This value is defined as the minimum number of links it is necessary to traverse, in order to reach one of the original target pages.

This context graph is used during a crawl by a set of Naïve Bayes classifiers which can then make an attempt to estimate the link distance from any generic page to a relevant page. This context awareness allows the crawler to continue on a crawl path even if the reward for following a link is not immediate, but several links away. However, links which are expected to lead more quickly to relevant pages are favoured by the crawler. A major limitation of this approach is the reliance upon the provision of sufficient backlinks to the seed set of target pages by an existing search engine.

Reinforcement Learning

Reinforcement learning is the name given to a machine learning framework which promotes optimal sequential decision making using rewards and punishments [Kaelbling et al. 96]. In reinforcement learning approaches to web crawling, the actions that generate a benefit during a web crawl are mapped. As a crawl progresses, more and more actions and their subsequent outcomes are added to the map. The map thus becomes influential in predicting which actions the system should undertake to achieve the greatest benefit.

In a focused crawling scenario the system identifies patterns of text in, and in close proximity to, hyperlinks. The text analysed includes the headers and titles of the page, in addition to the anchor text and text in close proximity to the anchor. The relevancy of the content linked to by previously encountered hyperlinks containing similar patterns can be used to determine which links are most likely to lead to relevant pages. An evaluation of this approach [Rennie & McCallum 99] produced impressive results. The evaluation used the crawler to harvest research papers from four Computer Science department web sites, in addition to pages relating to the officers and directors of 26 companies, from their websites. For each crawl training sets must be generated in advance and both transition and reward functions defined. These functions are used to calculate a value for each hyperlink in the collection which quantifies the benefit of following that hyperlink.

Intelligent Crawling

An "Intelligent Crawling" framework has been proposed [Aggarwal et al. 01], in which a classifier is trained as the crawl progresses. The aim of this approach is to statistically learn the WWW's link structure while performing a search. For each crawl or "resource discovery", a set of configurable "predicates" are defined, and subsequently used to estimate the probability that an unseen page will satisfy the information need. The predicates can be simple keywords, topical searches using hypertext classifiers, page-to-page similarity measures, topical linkage queries or any combination of these factors.

In intelligent crawling no linkage structure of the WWW, such as topical locality, is assumed. Instead, the crawler is tasked with gradually "learning" the linkage structure of the portion of the WWW it is traversing. A key assumption of this approach is that different features of the WWW will be more useful in assessing the relevance of any given page depending on the predicates defined to guide the crawl. For some predicates the unseen page's URL may be most valuable in predicting relevance, while for others it may be most worthwhile to examine the content of the linking page. The crawler is expected to discern such feature values over the course of a crawl.

The crawler is not seeded [Aggarwal et al. 01] like other focused crawling techniques, rather it begins at general points on the web and gradually begins to auto-focus as it encounters content which satisfies the pre-defined predicates. The crawler initially behaves like a general

purpose crawler, following all the links it encounters, but gradually it becomes more adept at selecting links which are more likely to lead to pages which are relevant to the user-specified predicate. However, the selection and refinement of the predicates which drive the entire crawl could be quite a complex task [Micarelli & Gasparetti 07]. It would be difficult for a non-technical educator to describe an entire educational subject area using such predicates.

Genetic Algorithm-Based Crawling

Genetic algorithms are inspired by the principles of Evolution and Heredity. In evolutionary biology, populations of species evolve with particular chromosomes and genetic structures. The species which are most suited to their environment survive longer and thus have an improved chance of reproducing and spreading their chromosomes. Species which are ill-suited to their environment tend to die out. This process is also known as "natural selection". Similarly, when applied to Information Retrieval, genetic algorithms employ a number of potential solutions which "evolve" through a set of operators such as inheritance, random mutation, crossover etc. The solutions which perform most effectively are retained while the in-effective methods are discarded.

InfoSpiders [Menczer & Belew 00], utilises a collection of autonomous goal-driven crawlers without global control or state in the style of genetic algorithms. This system is also known as ARACHNID which stands for Adaptive Retrieval Agents Choosing Heuristic Neighbourhoods for Information Discovery. These evolving crawlers or "agents" traverse the WWW in response to user-defined queries, attempting to mimic the behaviour of a human surfing the web. Each agent autonomously assesses the relevance of any given page to the query and calculates the most desirable crawl path to follow.

The user initially provides a list of keywords and a list of seed URLs. The collection of crawlers is initialised by pre-fetching each of these seed pages. Each agent is randomly allotted one of the seed URLs as its start point and given a random behaviour and an initial supply of "energy". The energy that a particular agent has dictates its survival. Agents are awarded energy if a crawled page appears to be relevant and are deducted energy for all network loads incurred. Thus, if an agent does not regularly reach pages which are deemed to be relevant, it will run out of energy and "die".

Each agent analyses the content of its seed page and uses this analysis to determine the relevance of the sites linked to from the seed. Based on these estimates the agent creates a crawl order. An agent can modify its crawling behaviour based on previous results, by learning which links are most likely to lead to relevant pages. Agents can be selected for reproduction. When two agents reach a page at the same time, they can be combined to produce offspring. Random mutation can also occur, which helps prevent agents from converging on sub-optimal solutions.

A key element of this approach is the ability for the user to provide optional relevance feedback. The user can assess the relevance of the pages visited by the crawler. These relevance assessments can be made during the course of a crawl and alter the subsequent behaviour of an agent. The process is described as being "akin to the replenishment of environmental resources; the user interacts with the environment to bias the search process" [Menczer & Belew 00].

However, despite its unique approach to crawling, an extensive study [Menczer et al. 01] which compared three focused crawling techniques, Best-first, PageRank and InfoSpiders, found the Best-first approach to show the highest harvest rate.

Another crawler based upon a genetic algorithm approach is the Itsy Bitsy Spider [Chen et al. 98]. This follows a similar process to InfoSpiders with some functional variations. The termination of an agent occurs when two consecutive generations do not produce improved relevancy with regard to retrieved content. There is no notion of rewards and penalties for choosing relevant content. There are similar evolutionary operators which perform crossover reproduction and mutation to evolve the agents. Mutation is achieved by utilising Yahoo's web directory to suggest possible promising new seed sites to individual agents. A Jaccard fitness function is used to assess the relevance of discovered pages. Each page is represented as a weighted vector of keywords and compared to the corresponding vectors for the seed set to determine the probability of it being relevant.

However, despite its unique approach to focused crawling, genetic algorithms have not fared well during evaluation. An extensive study [Menczer et al. 01] which compared three focused crawling techniques, Best-first, PageRank and InfoSpiders, found the Best-first approach to

show the highest harvest rate. During an evaluation of the Itsy Bitsy Spider approach [Chen et al. 98] the genetic algorithm approach again fails to improve upon the performance of a best-first search system. The recall values of the genetic algorithm approach are significantly better than the best-first system, but precision, which is most important in terms of the WWW, is not significantly improved.

Social or Ant-Based Crawling

Research into social insect collective behaviour has inspired a different approach to focused crawling [Gasparetti & Micarelli 03] [Gasparetti & Micarelli 04]. This model for focused crawling is based on how insects, in particular ants, help each other to find the way by marking trails with a hormone that can be detected and followed. In a similar manner a collection of crawling agents traverse the web and mark paths to relevant pages for other agents to follow. The assumption is that these agents upon exploiting this trail can make further explorations and discover more distant, relevant material than would have been otherwise possible.

This crawler also addresses the concept of context paths [Mizuuchi & Tajima 99]. Individual web pages are often not self contained (although as the navigation methods employed by users evolve, this is now less frequently true; see section 2.3.4). The author of a page sometimes assumes that the browser has a certain degree of knowledge in a subject area or has browsed through other sites to get to this page. Sometimes in order to satisfy an individual's information need, it is necessary to provide information from a variety of sources. These pages, when connected, can be presented as a single unit of information, or a context path.

Each path that is followed by a crawling agent is marked by a trail. Each trail is rated, or assigned a "pheromone intensity", dependant on the number of relevant resources found on a path, and the distance the crawler must travel to reach these resources. Each crawl is cyclical and a drawback to this approach is that the crawler must begin each cycle at the same collection of seed pages. This can lead to a large amount of duplicate crawling of pages and a subsequent waste of computational resources.

Analysis

Generic web crawlers, and the search engines based upon them, can be compared to public libraries: they try to cater for the public at large, with a supply of material to meet the majority of common information needs and preferences. Such systems do not specialize in specific areas of interest. This creates a problem when attempting to apply such technologies within the education domain. In TEL, the content required for the generation of educational experiences is often highly specialised and must meet the requirements of both the educator and curriculum.

Consider a Zoologist, highly experienced in her field, who delivers lectures in a very specific subject area, such as the Yangtze River Dolphin. The exponential growth of the web matters little to this educator if only 10 pages in her particular topic of interest are added or updated on a weekly basis. However, staying abreast of where this content resides, when and where new content appears and when content is updated is a non-trivial task which becomes ever increasingly difficult as the web expands. To locate such content, traditional web crawlers would waste huge amounts of computational resources traversing and indexing hundreds of millions of pages when the number of pages which are relevant and desired by the educator may be very small.

Now consider a very inexperienced learner who is only beginning to study a subject area. As discussed in the previous section, the WWW is vast in scale and offers information on a huge variety of topics, the boundaries between which can often be blurred. When searching for information, people attempt to avoid information overload by filtering the information with which they are presented both by relevance to their information need and by quality. For an inexperienced learner this can be a bewildering experience. They have very limited knowledge of the subject area in question and as such, their ability to judge relevance is impaired.

A system which can explore the WWW searching for sources of quality educational content on a particular subject and periodically re-crawl these sources for updates would be an invaluable tool. A suitably implemented focused crawling system could provide just such a service.

Focused crawling has been shown to be a powerful means for the discovery of topic-specific resources on the WWW. Such crawlers can be driven by a variety of different means including keyword descriptions or exemplar documents. Some crawlers also attempt to learn what constitutes a relevant document as a crawl progresses, through human-aided relevance feedback or machine learning. These crawlers then explore the Web, guided by relevance or popularity assessment mechanisms. Content is filtered at the point of discovery rather than post-indexing which improves its efficiency with regard to computational resources. Focused crawlers provide a means of exploring the WWW in a controlled, subject specific and yet dynamic fashion.

The numerous approaches to focused crawling described in the above section all use distinct techniques to discover relevant content. There are various aspects and limitations of each approach that should be kept in mind during the design of a focused crawling service for TEL.

When Link prioritisation is applied to subject-specific crawling, evaluations have shown breadth-first to be the most successful link-ordering algorithm. Taxonomic crawling could be used to allow a subject matter expert to define the crawl limits. However, this must be implemented in such a manner that it can be conducted by a non-technical user. The context graph approach is limited by it reliance upon existing search tools to backcrawl the web. In reinforcement learning, the definition of sets of transition and reward functions, which are necessary in this approach, could be difficult for a non-technical user. In intelligent crawling, the crawler starts by conducting a general purpose crawl and gradually focuses by following links which are more likely to lead to relevant content. These crawls will, by nature, be longer and more demanding on computational resources. The selection and refinement of the predicates which drive the entire crawl could also be quite complex. It would be difficult for a non-technical educator to describe an entire educational subject area using such predicates. When evaluated for focused crawling, genetic algorithm techniques fail to improve upon standard, best-first techniques. Social or Ant-based crawling techniques tend to result in large amounts of duplicate crawling and unnecessary resource consumption.

Summary

This section detailed the current trends and approaches to the implementation of focused or topic-specific crawling on the WWW. The section examined the underlying theory of topic

locality and how it relates to focused crawling. It then went on to analyse in turn the various approaches to performing the focussing, or relevance assessment, aspect of a web crawl. In the following section various open source implementations of web crawlers will be examined.

Open Source Web Crawlers

It is an goal of this research to utilise open source software, where possible, in an effort to avoid "recreating the wheel". There exist a number of web crawling systems which are open source and available for integration into individual research projects. All boast diverse and varying feature sets that satisfy the requirements of specific online communities. An objective of this research is to identify IR techniques and technologies which can be successfully applied within TEL. Conducting an analysis of the web crawlers most applicable to this research which are in active use was an essential precursor to the design of a focused content retrieval mechanism for this research. These web crawling tools are discussed in the section below.

Swish-e

Simple Web Indexing System for Humans – Enhanced (Swish-e) [Swish-e] is a free, open-source web crawler and indexing system. It was developed by the UC Berkeley Library in 1996, building upon the Swish web crawler. Swish was created in 1995 at Enterprise Information Technologies (EIT) by Kevin Hughes [Rabinowitz 03]. Swish-e can be used to crawl web sites, text files, mailing list archives and relational databases. Swish-e is well documented and undergoes active development and bug fixing. There is a lively mailing list that addresses issues and bugs in the system and receives regular input and feedback from project members as well as experienced users. The system also provides indexing functionality which is discussed in more detail in section 1.8.5.

Swish-e has two drawbacks that could impede its effectiveness at content analysis and retrieval on the open WWW. Firstly, the crawler does not provide multibyte support. This means that the crawler can only process 8-bit ASCII code and does not support 8-bit UCS/Unicode Transformation Format (UTF-8). UTF-8 is a variable length character encoding format for Unicode. Secondly, Swish-e is designed, and functions most efficiently, on small to medium sized data collections of less than a million documents. This does not provide sufficient scalability to conduct crawls on the open WWW.

i-Via Nalanda

The Nalanda focused web crawler [Nalanda] is an open source crawler developed by IIT Bombay and the U.S. Institute of Museum and Library Services for the iVia Virtual Library System project. It is based on the research conducted in [Chakrabarti et al. 99] [Chakrabarti et al. 02]. The crawler supports focused crawling on a predefined subject domain and employs two methods of classification to improve the accuracy of content topic matching.

Nalanda utilises the fact that resources focused around a common topic often cite one another, as discussed in sections 3.4.1 and 3.6.1. Highly inter-linked resources are examined, evaluated and rated for their capacities as authoritative information sources. These content sources are split into authoritative resources and hubs. The second method of classification that Nalanda employs is a keyword and vocabulary comparison between candidate resources and resources that have already been accepted into the collection. This classification occurs after the linkage analysis has been performed.

The crawler has been improved with the addition of an 'apprentice' component. The apprentice is a learning algorithm which intelligently recognises clues in hyperlinks to try and decide upon the most promising links to follow for the purpose of each crawl.

Combine Harvesting Robot

Combine [Combine] is an open source system for crawling, harvesting and indexing internet resources. It was initially developed as a general purpose crawler as part of the Development of a European Service for Information on Research and Education (DESIRE) project [Desire]. It was later modified for use as a focused crawler by the EU project ALVIS – Superpeer Semantic Search Engine [Ardö 05].

The general purpose web crawler has been combined with an automated subject classifier created by the KnowLib research group [KnowLib] to generate topic-specific databases of crawled information. The crawl focus is provided by the use of an ontology that is used for topic definition and term matching. When a document have been deemed relevant, further processing, such as character set normalization, language identification and simple text segmentation, is performed in preparation for processing of the data.

All crawled information is stored locally in a relational database. By using this central database for synchronisation, it is possible to run several crawlers in parallel, all crawling a single subject domain. Combine can be used for metadata extraction. It can handle multiple document types including plain text, HTML, PDF, PostScript, MsWord, LaTeX and images. A SQL database is used for central data storage and administration.

Before a crawl can be conducted the user must create what is termed a topic definition. The topic definition breaks each subject down into hierarchical subject classes. A file is then created by the user which contains keywords, phrases or boolean expressions which are related to the subject area. Each of these items must be associated with a subject class and have a term weighting applied. Term weights can be positive or negative. A list of seed URLs must also be collated and provided to the crawler as start points for the crawl.

Nutch

Nutch is divided into two distinct components: the web crawler and the searcher. The web crawler fetches web pages and creates an index. The searcher identifies content relevant to a user's search query. The crawler component of Nutch consists of three main components: a fetcher for content harvesting and link extraction; a custom database that stores URLs and the content of retrieved pages; and an indexer, which processes each page and builds a keyword-based index. The indexing functionality of Nutch will be discussed in more detail in section 3.8.4. Nutch is implemented in Java and is designed to support three distinct scales of crawling: local filesystem crawling; intranet crawling; and open web crawling.

The web database, or WebDB, mirrors the structure and properties of the portion of the WWW being crawled. Nutch defines this portion of the WWW as a web graph, where the nodes are web pages and the edges are hyperlinks. The WebDB is used only by the Nutch crawler and does not play any role during searching. Nutch groups crawls into segments. Each segment consists of the collection of pages fetched by the crawler in a single crawl instance. Each segment has a fetchlist, which is a list of URLs for the crawler to fetch. This is generated from the WebDB. The fetcher step then requests web pages, parses them, and extracts links from them. Both the content of pages and URLs are stored in the database. Numerous pieces of information about each page are stored, such as: the number of hyperlinks in the page; fetch information; and the page's score, which can be calculated using

selected link analysis algorithms. Nutch observes the Robots Exclusion Protocol when crawling the open web.

Heritrix

Heritrix [Mohr et al. 2004] is the Internet Archive's [Internet Archive] open source, extensible, web-scale, archival quality web crawler and is available under the GNU Lesser General Public License [LGPL]. The crawler was initially developed in 2003 and has since become a stable platform with a large community of users and developers that perform regular bug fixes and provide assistance through a lively mailing list. The crawler is implemented in Java and is designed to be able to perform *broad crawling*, *focused crawling*, *continuous crawling* and *experimental crawling*.

The crawler follows a standard crawling methodology using a pluggable architecture. A URI is chosen from among those scheduled and the content is fetched. The content is then either archived or analysed depending upon the purpose of the crawl. Any URIs in the content are extracted from the retrieved content and added to the queue. The URI is then marked as complete and the process is repeated recursively.

The Heritrix architecture is divided into three main components. The *scope*, which defines which URIs should be crawled. The *frontier*, which monitors the queue of URIs and schedules those to be downloaded. The final component is the *processor chain* which is composed of pluggable components which each perform an operation upon the content. These components can perform either pre or post-processing tasks or content analysis tasks.

An example of a system that used Heritrix to perform focused crawling tasks was Metacombine [Metacombine]. Metacombine was a Mellon Foundation-funded project hosted at Emory University to research methods to more meaningfully combine digital library resources and services. This project worked to combine metadata technologies such as OAI-PMH with web crawling techniques in an effort to improve scholarly communication. Heritrix was used to generate collections of web resources relating to the American south for use in the project.

Summary

In the previous sections IR on the WWW as a discipline was introduced. The emergence of specific techniques and methods in the area of web crawling were detailed. Focused crawling was then comprehensively examined as a means of traversing the web in a topic-specific fashion. In this section four publically available web crawling solutions were analysed and detailed as a precursor to the creation of a focused content retrieval mechanism for this research. The next section of this chapter will address post-crawl content discoverability issues, such as indexing and searching.

Indexing, Searching and Retrieving Content Sourced from the WWW

The representation of the content contained within documents and, in the case of Information Retrieval on the WWW, web pages is referred to as indexing. Indexing is a process which translates natural language content into a machine usable form. This involves methods of deducing which lexical terms best describe the content of a document. The index which is generated by this process is then used in combination with a user query, by a search engine or other such service, to locate and rank relevant information [Salton & McGill 84] [van Rijsbergen 79].

Indexing

The explosive growth of the WWW, as discussed in section 3.5, has made the user-driven discoverability of content, and consequently the indexing of content, ever increasingly important. The indexing mechanism is a critical component of any web-based IR system. It must provide a formalised, simplified and machine usable representation of the natural language content contained within each web page [Salton 89].

In traditional IR approaches to content indexing, each document is treated as an unstructured, unordered bag of words. Indexing is based upon the assumption that the occurrence of terms within a document can be used to determine the subject matter of the content. Pre-processing steps are conducted to simplify this term-based analysis. Pre-processing consists of a series of steps aimed at removing all information un-related to the semantics of the content. Once this is complete, terms which are deemed meaningful are extracted from the document and weights are calculated for each. These weights are used to signify the importance of the term as an indicator of the documents subject matter. This allows the IR system to discriminate

between documents with respect to terms in a user query, rank the documents according to relevance and present the most relevant documents to the user.

The following sections will detail the most common approaches to indexing in the IR field and subsequently, how these approaches have been adapted to deal with the issues and challenges of representing web content. The typical pre-processing conducted on pages will be explained followed by the some of the widely applied indexing, or document modelling, techniques.

Term Normalisation

In traditional IR approaches, term normalisation occurs as one of the document pre-processing tasks. Term normalisation refers to the conversion of a page form a structured written work, into an un-structured stream of text. This means the removal of all text cases and punctuation. The following is an extract from chapter seven of "Alice's Adventures in Wonderland" by Lewis Carroll:

Twinkle, twinkle, little bat.
How I wonder what you're at!
Up above the world you fly.
Like a tea-tray in the sky.

When term normalisation has been applied, this text would become:

twinkle twinkle little bat
how i wonder what you re at
up above the world you fly
like a tea tray in the sky

Stopword Removal

Not all terms aid discrimination between pages when searching an index. Some terms may be descriptive of the subject matter of an individual page, but occur so regularly in a collection of documents that they are poor discriminators between pages. Words are not evenly distributed across languages, be they in written or spoken form. Few words appear very

frequently, while many words appear very infrequently. This distribution of words is known as Zipf's distribution [Zipf 35] [Zipf 49].

Zipf's law states "in a corpus of natural language utterances, the frequency of any word is roughly inversely proportional to its rank in the frequency table". The most frequently occurring word will occur twice as often as the second most frequently occurring, which will occur twice as often as the fourth most frequently occurring and so forth. According to Zipf's law, when attempting to differentiate documents in a corpus, high frequency words hold little value. They describe too many individual pages in the corpus and add no unique descriptive value. Zipf's law also states that words which occur extremely infrequently in a corpus may also be of little value. They may be spelling mistakes or uncommon proper nouns. These words are too rare to be of value. The most valuable terms for discriminating between documents within a corpus are those which occur with medium frequency.

Frequently occurring terms, particularly connectives (i.e. and, but, because), articles (i.e. the, an, a) and prepositions (i.e. on, of, as, not), are commonly removed from an IR index. These terms are known as stopwords. Stopwords can be defined as any terms which add no, or limited, value to an index. This process reduces the number of common terms between pages in a collection. When stopword removal has been applied, the example text defined above would appear as follows:

twinkle twinkle little bat
wonder
above world fly
like tea tray sky

However, stopword removal can cause problems with page recall and the purpose of the index should be carefully considered before removing terms. Consider the Shakespearean text "to be or not to be", each of these individual terms hold no descriptive value and would generally be removed as stopwords. However when combined they form a well known quotation which is likely to be used as a search query against such Shakespearean texts.

Stemming

The process of stemming conflates morphologically similar terms to their morphological root, so that the index can recognise variations of the word when conducting a search [Lovins 68] [Porter 80] [Sparck Jones & Willet 97]. Words can occur in various forms and all variations need to be recognised as referring to a common concept. This process can aid the number of relevant pages returned for a search. Consider two pages on the WWW with the titles:

"The Best Hill Walking in Scotland"
"John Cleare's 50 Best Hill Walks"

Without stemming, if a user performed a search for "Hill Walking" only the first page would be returned. However stemming removes suffixes and would reduce both Walking and Walks to Walk. If a user then performed a search for "Hill Walking" the query would be stemmed to "Hill Walk" and both pages returned. A typical stemmer consists of a collection of rules and dictionaries [Krovetz 93]. These rules are specific to each language used and can be extremely complex. Care should be taken when defining such rules as there can be problems with the stemming process. Terms can be produced which are not actual words and the process can be difficult for a user to understand. For example, when using the Porter Stemming Algorithm [Porter 80], the following conflations occur:

computer	→	comput
ponies	→	poni
iteration	→	iter

Various errors can occur within the stemming process. For instance, connections between terms can be missed by the stemming algorithm:

european	−	europe

Unrelated terms can be falsely connected:

policy	→	police

Proper nouns are not recognised and can be stemmed in error:

thomas → Thoma

It is important to have a balanced approach to stemming so that the majority of the benefits of the process are reaped without unduly hampering retrieval performance.

Too Aggressive / Overstemming			**Too Timid / Understemming**		
Conflated			Not Conflated		
Same Index Term			Different Index Terms		
organisation	→	organ	european	→	europe
policy	→	police	cylindrical	→	cylinder
executive	→	execute	create	→	creation
army	→	arm	search	→	searcher

The majority of web search systems employ stemming. If a user searches for "*pet lemur dietary needs*", the search engine will also conduct searches for "*pet lemur diet needs*" and other related variations of the terms[13]. The stemming process also has the desirable side effect of reducing the index size. The process of stemming increases the number of common terms between resources in a collection. Individual resources containing the terms climbing, climber, climbs and climbed are now all described by the common term climb. This improves the recall of IR systems. Stop-word removal on the other hand, reduces the number of common terms between resources in a collection. This aids discrimination between resources and improves precision.

Part-of-Speech Tagging

Part-of-Speech analysis considers the role of individual words in text [Klein & Simmons 63] [Green & Rubin 71]. The same word can belong to different syntactic categories when used in different contexts. For example "He *books* tickets" as opposed to "He reads *books*". Tagging such syntactic roles can enable more complex forms of content analysis and

[13] An example provided by the Google Web Search Help Centre. For more information see http://www.google.com/support/?ctx=web

information extraction [Charniak 97]. Part-of-Speech tags are generally assigned from a standard set.

Tag	Description	Example
NN	Noun Singular	Table
NNS	Noun Plural	Tables
NNP	Proper Noun	John
JJ	Adjective	Green
VB	Verb Base Form	Take
VBZ	Verb, Present, 3rd Person	Takes
VBD	Verb Past Tense	Took
RB	Adverb	Here, However
PRP	Personal Pronoun	He, It

Figure 3-9. A sample of Part-of-Speech Tag set. The full list is available in [Manning & Schütze 99].

These tags are applied to individual words in each page. If applied to the example defined above, the following tags would be used:

"He books tickets"	→	He/PRP books/VBZ tickets/NNS
"He reads books"	→	He/PRP reads/VBZ books/NNS

Part-of-Speech tagging can also be used to identify phrases and known entities. Common phrases in a particular corpus, such as "United States of America" and "Natural Language Processing", can be tagged to aid the indexing process. Named entities can also be tagged, such as "George W. Bush" and "CIA".

Term Weighting

Early IR systems merely recorded the presence or absence of a term. This is known as binary retrieval. More advanced IR systems weight terms according to their relative estimated importance. This importance measure can be derived from a terms importance within a corpus of content or a terms importance within and individual page. These measures can be influenced by each other. If a term occurs frequently within a corpus, its importance within

an individual page is less significant. For instance, the terms "Dublin" and "Traffic" occurring frequently in a page are of little importance if the corpus being indexed is a collection of Dublin Transport Office web pages.

Term frequency (*tf*) is used to measure the importance of a term in an individual document. *tf* is defined as:

$$tf_{dt} = \left(\frac{num_t}{total_d}\right)$$

Where $total_d$ is the total number of terms in a document d and num_t is the total number of times that a term t occurs in d. *tf* returns high values for frequently occurring terms. Tf is calculated for all index terms on all documents in the corpus. A Term-Document Frequency Matrix can then be constructed for the collection. This consists of all the index terms of all the documents in the corpus.

	d_1	d_2	...	d_n
t_1	tf_{11}	tf_{12}	...	tf_{1n}
t_2	tf_{21}	tf_{22}	...	tf_{2n}
...	...	...	...	...
t_m	tf_{m1}	tf_{m2}	...	tf_{mn}

Figure 3-10. Term-Document Frequency Matrix.

Inverse document frequency (*idf*) is used to measure the importance of a term within a corpus of content [Sparck Jones 72]. *idf* is defined as:

$$idf_t = \log\left(\frac{n}{n_t}\right)$$

Where t is a term in a document collection n. n is the number of documents in the collection and n_t is the number of documents in the collection in which t occurs. *idf* returns high values for infrequently occurring terms.

Term frequency and inverse document frequency are often combined in IR to provide a more accurate overall term weight (*tf-idf*) [Salton & Yang 73] [Salton & Buckley 88].

$$weight_{dt} = idf_t * tf_{dt}$$

A high *tf-idf* value is achieved if a term has a high frequency within the document in question, but a low frequency in the corpus as a whole. A proposal has been made to adapt the *tf-idf* method and apply it to the anchor text of hyperlinks on the WWW [Hawking et al. 04].

Other weighting schemes have been developed with particular IR models in mind. For instance, a method based on genetic programming has been defined [Cummins & O'Riordan 06] which can automatically determine term weighting schemes for the Vector Space Model (VSM). The VSM will be discussed in more detail in section 3.8.2.2. This is based on a set group of queries and a user-selected collection of relevant documents for the queries. Weighting schemes are then evolved which achieve a high level of retrieval precision on the corpus. Schemes are evolved in document specific (local) and corpus specific (global) domains and combined to produce the best results.

Analysis

Term frequency and term weighting techniques will be particularly important when creating caches of subject specific content for use in educational scenarios. As a collection of content will generally be on a single subject, some terms which are very valuable during a web crawl, will be less valuable during indexing. For instance, if collecting content related to "The Poetry of W.B. Yeates", the term "Yeates" will be extremely useful when searching for relevant content on the open WWW during a crawl. However, when later searching across a collection of content consisting solely of pages related to Yeates' poetry, the value of the term "Yeates" may be greatly reduced.

Summary

Indexing is the process of representing content so that an IR system can discriminate between resources with respect to a user's information need. The IR system uses the index to make a relevance assessment and present the most relevant resources to the user. This section has provided the reader with an introduction to the most successful approaches to indexing in the IR field. Content pre-processing is examined and the processes of normalisation, stopword removal and stemming are explained. This is followed by an introduction to Part-of-Speech tagging techniques. Finally the section concludes with an detailed examination of approaches

to index term weighting. In the next section contrasting computational models for IR are detailed. These include the Boolean model, Vector Space model and Probabilistic model.

Information Retrieval Models

IR models form a description of the computational process of retrieval. This includes the process by which an information need is initially articulated and possibly refined. The model also defines the method of selecting a document from among a collection for retrieval. The IR models examined here are the most popularly applied models. Each is based upon user generated queries which consist of one or more keywords. The semantics of the resources in a collection and the user information need can both be expressed through combinations of such terms. The relevance of a resource can then be estimated using similarity functions which compare the information need and the available resources.

Boolean Model

The Boolean Model approach to IR is based upon Boolean Logic, Boolean Algebra and Set Theory. It is one of the oldest IR models. In this approach, pages are represented as a set of keywords which have been automatically or manually extracted from all the pages in a collection [Salton & McGill 83] [van Rijsbergen 79]. Each page in the collection is represented as a vector:

$$\overrightarrow{d_p} = \{(t_1, w_{1p}), (t_2, w_{2p}), \ldots\ldots, (t_n, w_{np})\}$$

A binary value is assigned as the weight w, of a term t, based upon its occurrence or non-occurrence in a document d. The weight is set to 1 if the term occurs in the document in question and the 0 if it does not occur.

In the Boolean Model each query is represented as a Boolean expression or terms and their connectors (AND, OR etc.). These connectors are usually expressed as INFIX operators:

$$((a\ AND\ b)\ OR\ (c\ AND\ b))$$

The NOT clause is a PREFIX operator:

$$(c\ AND\ (NOT\ (b)))$$

The AND and OR clauses are n-ary:

$$(a\ AND\ b\ AND\ c \ldots)$$

Vector Space Model

A vector defines a position in space. In the Vector Space Model (VSM) approach to IR each document is represented as an n-dimensional vector. There is one dimension for each index term in the collection. Calculated term weights or Boolean values can be used. This approach was first implemented in the SMART retrieval system [Salton 71].

Each document in the collection undergoes all the pre-processing steps already described above. Once the pre-processing steps are complete, a Term-Document Frequency Matrix is generated for the collection. Consider the following example:

	d_1	d_2	d_3	d_4
t_1	1	1	2	1
t_2	3	0	1	1
t_3	0	1	3	0
t_4	1	2	0	2

Figure 3-11. Example Term-Document Frequency Matrix.

Generally the vector for each document is sparse in nature, i.e. it contains a very small subset of the full list of terms. Every term $t_i \in d_j$ has a weight w_{ij}, such that: $w_{ij} > 0$ *if* $t_i \in d_j$ and t_i does not belong to all documents in the corpus [Micarelli et al. 07]. These weights are calculated using some measure of relevance, such as *tf-idf*. When the weights are calculated the vectors can be defined as follows:

	w_2	w_3	w_4
$\overrightarrow{d_1}$	0.418	0.000	0.176
$\overrightarrow{d_2}$	0.000	0.176	0.298
$\overrightarrow{d_3}$	0.176	0.418	0.000
$\overrightarrow{d_4}$	0.176	0.000	0.298

Figure 3-11. Vector Table with tf-idf weights.

Weight w_1 was excluded from the vector generation as it occurred in all documents in the collection as displayed in the Term-Document Frequency Matrix. Each co-ordinate of the vector space corresponds to an index term in the corpus and its importance in terms of that document. Queries are also represented, using the same rules as documents, as an n-dimensional vector.

$$\vec{q} = \{w_{1q}, w_{2q}, \ldots, w_{nq}\}$$

Once a query has been represented as a vector in the same n-dimensional concept space as the document corpus, relevant documents can be retrieved using very simple similarity functions. Cosine Correlation is the most employed of these similarity functions.

$$sim(d_j, q) = \cos(\vec{d_j}, \vec{q}) = \frac{\vec{d_j} * \vec{q}}{|\vec{d_j}| \, |\vec{q}|}$$

This similarity function is equal to the cosine of the angle formed by the vectors $\vec{d_j}$ and $\vec{q}$.

Probabilistic Model

The Probabilistic Model attempts to estimate the probability that a user will find a given document relevant to their query. This captures the IR problem within a probabilistic framework [Robertson & Sparck Jones 76] [Sparck Jones et al. 00a] [Sparck Jones et al. 00b]. The Probabilistic Model employs binary weight vectors in a similar fashion to the Boolean Model. However the query-document similarity function in the Probabilistic Model is more complex.

$$sim(d_j, q) = \frac{P(R/d_j)}{P(\bar{R}/d_j)}$$

Where R is the set of relevant documents from the corpus and $\bar{R}$ is the set of non-relevant documents from the corpus. $P(R/d_j)$ is the probability that document d_j is relevant to the query q. $P(\bar{R}/d_j)$ is the probability that d_j is not relevant for q.

Bayes rule for converting conditional probabilities [Bayes 1763] states:

$$P(a \mid b) = \frac{P(b \mid a) \, P(a)}{P(b)}$$

Where a represents a specific hypothesis, in this case that b is relevant to a given query. $P(a)$ is called the prior probability of a, which is inferred before b is analysed. In other words $P(a)$ is the initial estimated probability that any given page will be relevant to the query without knowing anything about the page. $P(h \mid a)$ is the conditional probability of h, given a. $P(b)$ is called the prior or marginal probability of b, and acts as a normalising constant. $P(a \mid b)$ is called the posterior probability and is derived having seen b, based on the likelihood of b occurring where a does or does not hold true.

The odds of an event happening are also often considered in the Probabilistic Model. The odds of an event a occurring are defined as:

$$O(a) = \frac{P(a)}{P(\overline{a})} = \frac{P(a)}{1 - P(a)}$$

By applying Bayes rule and odds rather than probabilities [Micarelli et al. 07], it is possible to expand upon the original query-document similarity function as follows:

$$sim(d_j, q) = \frac{P(d_j/R)\, P(R)}{P(d_j/\overline{R})\, P(\overline{R})}$$

Assuming that $P(R)$ and $P(\overline{R})$ are constant for each document in the corpus, this can be reduced to:

$$sim(d_j, q) \sim \frac{P(d_j/R)}{P(d_j/\overline{R})}$$

Assuming that all terms in the corpus occur independently and by switching to logarithms, this query-document similarity function can be written as:

$$sim(d_j, q) \sim \sum_{i=1}^{n} log \frac{P(t_i/R)\; P(\overline{t_i}/\overline{R})}{P(t_i/\overline{R})\; P(\overline{t_i}/R)}$$

This is based upon the previous assumption that d_j is a vector, consisting of n independent binary index term occurrences:

$$P(d_j/R) = \prod_{i=1}^{n} P(t_i/R)$$

Where $P(t_i/R)$ is the probability that a term t_i occurs in a document randomly selected from the corpus R, and $P(\overline{t_i}/R)$ is the probability that the term t_i is not present in a document

randomly selected from R. However, this equation is not valid initially as there are no retrieved documents upon which to base the probabilities. In this case simplifying probabilities are applied to the formula as follows:

$$P(t_i/R) = 0.5$$

$$P(t_i/\overline{R}) = n_i/N$$

Where n_i is the number of documents in the corpus in which t_i occurs and N is the total number of documents in the corpus. Based upon these assumptions the final calculation formula for term probabilities can be defined as:

$$P(t_i/R) = \frac{V_i + \frac{n_i}{N}}{V + 1}$$

$$P(t_i/\overline{R}) = \frac{n_i - V_i + \frac{n_i}{N}}{N - V + 1}$$

Where and N is the total number of documents in the corpus, n_i is the number of documents in the collection in which t_i occurs, V is the overall number of documents currently retrieved and V_i is the number of retrieved documents which contain the term t_i.

Language Model

The Language Model approach to IR estimates query-document similarity in a different manner to the other mainstream IR models. When searching for information, users commonly form queries consisting of terms which they expect to appear in relevant documents or web pages. The Language Model attempts to model this process. When searching across a collection of documents, an individual document is considered relevant if modelling that document is likely to produce the query in question. Unlike the Probabilistic Model which calculates the probability that a document is relevant to a given query, the Language Model approach builds a probabilistic language model for each document in the collection. Documents are then ranked based upon the probability of each model generating the given query [Ponte & Croft 98] as shown below:

$$P(Q|D) = P(q_1, q_2, \ldots, q_n|D) = \prod_{i=1}^{n} P(q_i|D)$$

The last term is obtained from the assumption of conditional independence of terms given the document's language model. Since a document or web page is typically larger than a keyword query, estimating a language model for a document is a simpler task than estimating

a model of relevant documents based on a query. Thus, the Language Model approach avoids the problem encountered by the Probabilistic Model of generating initial estimated assumptions of relevancy.

Each document's model D can be considered as a unique class and the task of the query-document similarity function is to classify a query into its best class, as given by the posterior $P(D|Q)$. Then by applying Bayes' rule, the function can be estimated as follows. If we assume a uniform prior $P(D)$ over all documents, then the posterior depends entirely on the conditional $P(Q|D)$ which is the query-document similarity function used by the Language Model.

The Language Model approach has similarities with traditional tf-idf methods. Term frequency is directly represented in tf-idf, and much recent work has defined the importance of document length normalisation [Karbasi & Boughanem 06]. The combination of document generation probability with collection classification acts in a similar fashion to idf. Terms which are rare in the document collection but common in individual documents will be more influential when estimating relevancy. Both models also share the assumption of term independence. However, in terms of performance, the Language Model significantly outperforms pure tf-idf measures [Manning et al. 08], and has typically equalled the performance of the Vector Space Model [Dai et al. 05].

Web-based Indexing and Retrieval

When applied to the WWW [Agosti & Melucci 00] [Kobayashi & Takeda 00], the traditional approaches to IR and indexing discussed above can be further refined to account for the structure of web pages. Using the tag structure of HTML, and similar XML-based content formats, it is possible for term weights to be altered based upon where the terms occur within the page. For instance, a term which occurs once in the title of the page is much more indicative of the subject matter than a term occurring multiple times in a paragraph further down the page.

This approach can be extended to extract distinct semantics from the page structure or to deduce term importance based on the physical characteristics of a page. Take for example scholarly research papers on the web which are often organised into sections such as *Title, Authors, Introduction, References* etc. Information regarding the nature of the terms and how

they inform the subject matter of the entire page can be deduced from the section in which they appear [Cutler et al. 99] [Molinari et al. 03]. This takes into account the entire syntactic structure of a HTML page. Term weights can be calculated based upon the importance of the tags in which they appear in relation to the structure of the page. Traditional IR weighting methods, such as *tf-idf*, are used to calculate the importance of a term within a tag. These values are then aggregated according to the importance of each tag in the page. Example tag hierarchies appear as follows:

Rank	Class Name	TAG / Parameter
1	Title	TITLE, META Keyword
2	Header 1	H1, Font Size=7
3	Header 2	H2, Font Size=6
4	Header 3	H3, Font Size=5
5	Linking	A HREF
6	Emphasised	EM, STRONG, B, I, U, STRIKE, S, BLINK, ALT
7	Lists	UL, OL, DL, MENU, DIR
8	Emphasised 2	BLOCKQUOTE, CITE, BIG, PRE, CENTER, TH, TT
9	Header 4	H4, CAPTION, CENTER, Font Size=4
10	Header 5	H5, Font Size=3
11	Header 6	H6, Font Size=2
12	Delimeters	P, TD, Text not included within any Tag, Font Size=1

Figure 0-6 Tag Hierarchy for Term Weight Calculation [Molinari et al. 03] [Marques Pereira et al. 05].

Class Name	HTML Tags
Anchor	A
H1-H2	H1, H2
H3-H6	H3, H4, H5, H6
Strong	STRONG, B, EM, I, U, DL, OL, UL
Title	TITLE
Plain Text	None of the Above

Figure 0-7 Tag Hierarchy for Term Weight Calculation [Cutler et al. 99].

The Vector Space Model is used as the query-document similarity function. This approach has been extended [Marques Pereira et al. 05] to add a contextual element to the term

weighting process. The contextual aspect of the process allows for the consideration the style of HTML tag distribution in the documents of a corpus. The typographical appearance of the content within a web page can also be used to discern term significance within the page. Character dimension and emphasis, such as underlining and font style, can be used to influence term weighting [Bordogna & Pasi 00].

Some approaches have combined traditional IR indexing and retrieval methods with HTML tag-influenced weighting systems and link-analysis algorithms like PageRank and HITS, discussed previously in section 3.4.1. One particular fusion approach [Yang 01] combined web-specific weighting measures with a modified HITS algorithm. Index term weight was calculated based upon the HTML tag in which the term occurs. For instance, the frequency of a term is increased by a factor of ten if the term occurs in the HTML header. This approached is then combined, using the Similarity Merge method, with the link-analysis and ranking methods of HITS. However, the authors found that fusing this weighting method with HITS link-based retrieval was not beneficial. The best text-based retrieval system in the trial not only outperformed the best link-based system, but also outperformed all fusion approaches also.

Analysis

The Boolean Model approach is quite simplistic and very easily implemented. However it is also quite limited in scope. It will only retrieve exact matches, and does not estimate relevancy. Either the query terms that are submitted all occur within a page or it will not be presented as a possible result. All exact matches are presented in an unordered list. There is also no weighting of terms in relation to their importance in a corpus. All terms are deemed to be of equal importance, a regularly invalid assumption, as was already shown in previous sections.

The Vector Space Model can be used to perform more subtle measures of resource relevancy for a search query than the Boolean Model. Documents do not have to be exact matches to be retrieved and presented to the user. Term weighting is applied for each term in the search query and an n-dimensional vector is created for every resource in the collection, as well as for the query. Retrieved documents can also be ranked for relevancy, most commonly using cosine correlation.

In the Probabilistic Model, retrieved documents are ranked according to the probability that they are relevant to a given query. This approach assumes the division of documents into relevant and non-relevant sets. Binary weights are employed in a similar fashion to the Boolean model. The probabilistic model fails to take into account the frequency with which terms occur within a document or within a collection. The model also imposes some restrictive simplifying assumptions such as term and document independence.

The Language Model applies probabilistic approaches to IR but in a different manner to the Probabilistic Model. The Language Model is a generative model and attempts to model the manner by which users generate queries to retrieve content. Term frequency is accounted for in this approach and an initial model of relevancy does not need to be estimated, unlike the Probabilistic Model. Based on current evaluations, the performance of the Language Model appears to be generally equivalent to that of the Vector Space Model.

These traditional IR techniques can be further supplemented by exploiting the standardised structure of web documents. As described in the above section, this can be achieved by altering the term weighting process. Such a supplemental approach to content indexing and searching could prove very useful in the development of a tool to search collections of educational content sourced from the WWW. The vast majority of educational content encountered will be in a structured form such as HTML, XHTML, PHP or similar. This would enable a finer level of relevance assessment across the collection of content and thus improve the retrieval quality.

Summary

This section has provided the reader with a detailed overview of three contrasting computational models for IR. The most basic, and oldest, of the three models in discussed first, the Boolean model. This is an approach based on Boolean Logic, Boolean Algebra and Set Theory. The Vector Space model is then detailed. In this approach each content resource is represented as an n-dimensional vector. Relevance assessment is then conducted across the vector space. The section concludes by examining both the Probabilistic Model and the Language Model, which estimate relevancy through the calculation of probabilities. The next section examines various approaches to web-based search and the user interfaces types used by these tools.

Retrieval Interfaces

When a user executes a search on the WWW, they are driven by some form of information need [Schneiderman et al. 97]. However, the user's intent when searching is not always informational; it can also be navigational or transactional. Navigational searches occur when a user is aware a site exists and needs to find its URL. Transactional searches occur when a user needs to make a transaction, such as a goods purchase, and needs to find a site where they can perform this transaction [Broder 02]. Web IR Interfaces have evolved around satisfying these distinct information needs.

A diverse variety of interface types have become available as the number of web IR tools competing for traffic has grown. The majority of mainstream search engines are designed to satisfy information needs and employ a simple keyword-based search interface. Google [Google] has always been noted for its minimalist design which makes its interface very intuitive to use. Many other search engines have similar interfaces including AltaVista [AltaVista], Ask.com [Ask], Cuil [Cuil] and Bing [Bing]. Yahoo! [Yahoo!] has similar search functionality but couples this with a directory structure of the WWW which can be browsed. The Yahoo! homepage is much more cluttered as it acts as a portal to other functionality such as email and instant messaging. However, it does now have a dedicated search homepage which follows a minimalist design similar to Google's. Most mainstream search tools offer "advanced search" functionality which allows users to define more explicit queries. Parameters such as desired content format, the number of results returned and the language of content returned can be specified.

More recently, social search platforms such as Eurekster [Eurekster] have emerged. Social search allows the user to leverage the accumulated knowledge of communities via custom search portals called swickis. A user can build and customise a swicki on any topic. This portal can then be shared and distributed to grow a community of interested users. The swicki becomes more valuable and useful the more it is used. Yahoo! Answers [Yahoo! Answers] is another form of social search. It allows users to ask questions of the user base, post answers and browse popular question threads.

Spock.com [Spock] is a "people search" tool. It allows the user to use a keyword search input box to enter a name, email, location or tag to find information about individuals. Numerous

"meta search engines" have appeared which attempt to leverage the power of established, popular search tools by combining their result sets. Dogpile [Dogpile] combines the search result listings of Google, Yahoo!, MSN Live Search and Ask in one result set.

Another variation of WWW IR tool that has emerged are "vertical" or specialist search engines, where the interface is tailored toward a particular community. Yahoo! Kids [Yahoo! Kids] is a portal for children including basic keyword search and "Ask Earl" a form of tailored social search. Cranky.com [Cranky] describe their service as the first "age relevant" search engine and is targeted at the over 50's. TrueLocal [TrueLocal] and YourLocal [YourLocal] are transactional search tools which provide an interface where a user can specify a service and a location and receive a list of service providers at that location.

Other WWW IR tools are attempting to provide interfaces with innovative results visualisation methods. KartOO [KartOO] combines meta search functionality with an interactive Mindmap display of its results list. Snap [Snap] combines a traditional keyword search interface and ranked results listing with dynamic page preview functionality when a result is hovered over with the cursor. Tafiti [Tafiti] is an experimental search interface from Microsoft, powered by Silverlight, which employs various novel techniques in its interface, such as allowing the saving of individual results from a ranked list for future use. Tafiti also allows the display of results lists as branches of a tree which can be grown, shrunk or rotated.

Analysis

Despite the numerous search techniques and types of user interface available, various studies [Deepak & Parameswaran 05] [Bharat & Chang 03] [Quesenbery 03] have found that users prefer simplistic user interfaces when searching for information on the WWW. Users favour simple keyword based approaches where they can type in a query immediately and where query refinement is easy and quick. This is beneficial as a search proceeds through iterations or as a user's information need evolves. Notably, this approach is deemed more desirable than the use of pre-query formulation features to aid the production of queries which are more likely to retrieve accurate results. It was also found [Deepak & Parameswaran 05] that the "advanced search" functionality of web retrieval tools such as search engines are rarely used.

Indexing and Retrieval Software

As mentioned previously, this research aims, where possible, to utilise open source software solutions to address some of the requirements raised by this state of the art analysis. There are numerous open source indexing and retrieval software platforms available for use with collections of content sourced from the WWW. Conducting an analysis of some of the more successful solutions in active use was an essential precursor to identifying methods which could be applied within TEL, in design of a content retrieval mechanism. These open source indexing and retrieval tools are discussed in the section below.

ht://Dig

The ht://Dig [ht://Dig] system provides open source indexing and search functionality available under the GNU General Public Licence [GPL]. The system was developed at San Diego State University as a means of indexing and searching the content on various web servers on the college network. ht://Dig uses fuzzy word indexing algorithms to index term occurrence in documents based upon different criteria such as synonym rules and word ending rules. The system provides a Common Gateway Interface (CGI) program for conducting searches. This can be invoked via a HTML form in a web page. There is also a command line option. Searches are conducted using keywords which are matched to the terms in the index. This system is specifically designed to cover the search requirements of a single company, campus or web site. It does not support web-scale indexing and searching.

Lucene

Lucene [Lucene] is a full-text indexing and search library available under the Apache Software Licence [ASL]. The flagship Lucene product is coded in Java, however Lucene has also been ported to various other programming languages including Delphi, Perl, C#, C++, Python, Ruby and PHP. Lucene is a standalone library which provides functionality for indexing and search. However it does not support parsing for the individual document formats that may be encountered in a selection of content sourced from the WWW. A separate parser needs to be provided for each document format, such as HTML, PDF, PHP etc. Once the text within a document has been parsed, Lucene provides several text analysis tools to aid the indexing process. These analysers facilitate pre-processing steps such as text normalisation, stopword removal and stemming. Lucene creates indexes, which are split into segments, and support *tf-idf* style term weighting. Query-Document similarity in Lucene is conducted using a combination of the Vector Space Model and the Boolean Model. The

Boolean Model is used to initially filter the documents that are relevant based upon the use of boolean logic in the query analysis.

Nutch

Nutch [Nutch] is an open source search engine which provides both web crawling and web-based search functionality. The web crawling component of Nutch is described in more detail in section 3.7. The indexing and search functionality of Nutch is based upon Lucene. However, it supplements Lucene with the ability to add web-specific tools such as document parsers and a link-graph database. Once content is downloaded it is added to a database and link and anchor text analysis are conducted to form a link graph.

The content parsing and indexing functionality of Nutch is implemented almost entirely by plugins, and is not shipped as part of the core code. This means that for every content format that the crawler must be able to handle, a new plugin must be either found or authored. Parsing plugins are available for common document formats including HTML, PDF and DOC. Nutch also enables charset detection and processing.

The interface between the crawler and searcher components of Nutch is the index. These two components are designed to be highly decoupled. However, in practice this is not quite the case. As the content of each web page is not stored directly in the index, the searcher component requires access to the crawl segments stored in the WebDB in order to produce page summaries and to provide access to cached pages.

Lemur

Lemur [Lemur] is a toolkit for IR which provides indexing functionality based on the Language Model approach, although it can also be adapted to support other models such as VSM. Lemur is designed for a broad range of IR applications including ad hoc and distributed retrieval, cross-language IR, content filtering, and categorisation. The toolkit also provides basic search functionality using approaches such as Okapi [Karamuftuoglu et al. 02] and KL Divergence [Lafferty & Zhai 01]. Lemur is open source and the project actively encourage users to modify the toolkit in support of their own research.

Lemur currently supports a variety of indexing functionality including: English, Chinese and Arabic language text; word stemming (Porter and Krovetz); stopword omission; acronym

recognition; part-of-speech and named entity recognition. The Lemur toolkit provides a stand-alone GUI which provides keyword-based search. Lemur is implemented in C++, while the search interface is implemented in Java/Swing. It is compatible with UNIX, Linux and Windows.

Swish-e

Simple Web Indexing System for Humans – Enhanced (Swish-e) [Swish-e] [Rabinowitz 03] is a free, open source web crawler and indexing system. The web crawling functionality and history of the system has already been described in section 3.7.1. Swish-e is designed, and functions most efficiently, on small to medium sized data collections of less than a million documents. Using the GNOME [GNOME] libxml2 [libxml2] parser and a collection of filters, the system can index standard data formats such as HTML, XML, DOC, PDF, PS and PPT among others. Indexing can be limited to selected metadata tags or XML elements, sections of a web site, or to relevant pages in a web crawl using regular expressions to decide relevancy. The default Query-Document similarity ranking is conducted using basic *tf* with the ability to bias term weighting based upon where it occurs in the page. There is also the option to use a configurable *tf-idf* function that allows the limitation of term frequency analysis to particular sections of a document. Swish-e also enables the creation of experimental ranking schemes. A drawback of the indexing functionality is that it does not support the deletion of content from an Index. To remove a document, the entire index needs to be regenerated.

Xapian

Xapian [Xapian] is an open source indexing and search library, released under the GNU General Public Licence [GPL]. It is implemented in C++, but provides bindings which enable its use from systems written in Perl, Python, PHP, Java, Tcl, C# and Ruby. Xapian is a highly adaptable toolkit which allows developers to easily add advanced indexing and search facilities to their own applications. Query-Document similarity is based upon the Probabilistic Model using an implementation of the BM25 or Okapi Probabilistic weighting scheme [Sparck Jones et al. 00a] [Sparck Jones et al. 00b]. Xapian also allows the use of a Boolean weighting scheme or the specification of customised weighting schemes. The search interface supports a rich set of Boolean query operators. Omega [Omega] is an open source search interface which can be used in combination with Xapian.

WebGlimpse and Glimpse

WebGlimpse [WebGlimpse] is indexing and search software for WWW content which is free to use within the education, government and not-for-profit domains and available via a licence for commercial sites. It is based upon Harvest [Harvest] a distributed search engine framework originally developed in 1995 at the University of Arizona. The software has several independent components including a web crawler, implemented in Perl, and Glimpse, which is the core indexing and search functionality, implemented in C. Glimpse, which stands for GLobal IMPlicit Search, generates indexes which are then available through a simple CGI interface. The number of terms used in the generation of a Glimpse index is configurable to control index size. The WebGlimpse search interface supports Boolean termed search and pattern matching for relevance.

Summary

In the previous sections post-crawl content discoverability was analysed and the related challenges outlined. Indexing and retrieval were introduced as component disciplines of IR. The emergence of specific techniques and methods in the field were detailed. Indexing approaches and Query-Document similarity models were then comprehensively examined as a means of discovering relevant content from a document collection. Six publically available indexing and retrieval software solutions were analysed and detailed as a precursor to the creation of a content retrieval mechanism for this research.

Conclusions

With the continued growth of TEL in mainstream education, there is an ever-increasing demand for high-quality digital educational content resources. This creates a major problem for educational institutions, as the development of such resources is an extensive undertaking and an expensive process. The WWW is a significant source of digital content which remains largely unexploited in TEL due to problems with content discoverability and reuse. Techniques and technologies employed in Information Retrieval such as focused crawling, content indexing and retrieval models can be used to aid the discovery and aggregation of educational content sourced on the WWW. These techniques and technologies can be used in the implementation of a service which supports the discovery, classification, harvesting and delivery of educational content from open corpus sources, which is an aim of this research.

Appendix C – Order.xml

```
<?xml version="1.0" encoding="UTF-8" ?>
<crawl-order xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="heritrix_settings.xsd">
  <meta>
    <name>Seamus Lawless - PhD Experiment - Fifth Crawl</name>
    <description>Default Profile</description>
    <operator>Admin</operator>
    <organization>TCD</organization>
    <audience />
    <date>20070216180000</date>
  </meta>
  <controller>
    <string name="settings-directory">settings</string>
    <string name="disk-path" />
    <string name="logs-path">logs</string>
    <string name="checkpoints-path">checkpoints</string>
    <string name="state-path">state</string>
    <string name="scratch-path">scratch</string>
    <long name="max-bytes-download">0</long>
    <long name="max-document-download">100000</long>
    <long name="max-time-sec">0</long>
    <integer name="max-toe-threads">90</integer>
    <integer name="recorder-out-buffer-bytes">4096</integer>
    <integer name="recorder-in-buffer-bytes">65536</integer>
    <integer name="bdb-cache-percent">0</integer>
    <newObject name="scope" class="org.archive.crawler.scope.BroadScope">
      <boolean name="enabled">true</boolean>
      <string name="seedsfile">seeds.txt</string>
      <integer name="max-link-hops">25</integer>
      <integer name="max-trans-hops">5</integer>
      <newObject name="exclude-filter"
class="org.archive.crawler.filter.OrFilter">
        <boolean name="enabled">true</boolean>
        <boolean name="if-matches-return">true</boolean>
        <map name="filters">
        <newObject name="removeNonTextPages"
class="org.archive.crawler.filter.URIRegExpFilter">
          <boolean name="enabled">true</boolean>
          <boolean name="if-match-return">true</boolean>
          <string
name="regexp">.*(?i)\.(a|ai|aif|aifc|aiff|asc|bcpio|bin|bz2|c|cdf|cgi|cgm|c
lass|cpio|cpp?|cpt|csh|css|cxx|dcr|dif|dir|djv|djvu|dll|dmg|dms|dtd|dv|dvi|
dxr|eps|etx|exe|ez|gram|grxml|gtar|h|hdf|hqx|ice|ics|ief|ifb|iges|igs|iso|j
nlp|jp2|js|kar|lha|lzh|m3u|mac|man|mathml|me|mesh|mif|ms|msh|mxu|nc|o|oda|o
gg|pbm|pct|pdb|pgm|pgn|pl|pnm|pnt|pntg|ppm|py|qt|qti|qtif|ra|ram|ras|rdf|rg
b|rm|roff|rpm|rtx|s|sgm|sgml|sh|shar|silo|sit|skd|skm|skp|skt|smi|smil|snd|
so|spl|src|srpm|sv4cpio|sv4crc|swf|t|tar|tcl|tex|texi|texinfo|tgz|tr|tsv|us
tar|vcd|vrml|vxml|wav|wbmp|wbxml|wml|wmlc|wmls|wmlsc|wrl|xbm|xht|xpm|xsl|xs
lt|xwd|xyz|z|zip)$</string>
        </newObject>
        <newObject name="pathdepth"
class="org.archive.crawler.filter.PathDepthFilter">
          <boolean name="enabled">true</boolean>
          <integer name="max-path-depth">20</integer>
          <boolean name="path-less-or-equal-return">false</boolean>
        </newObject>
        <newObject name="pathologicalpath"
class="org.archive.crawler.filter.PathologicalPathFilter">
          <boolean name="enabled">true</boolean>
          <integer name="repetitions">3</integer>
        </newObject>
```

```
        </map>
      </newObject>
    </newObject>
    <map name="http-headers">
      <string name="user-agent">Mozilla/5.0 (compatible; heritrix/1.4.0
+http://www.cs.tcd.ie/seamus.lawless)</string>
      <string name="from">seamus.lawless@cs.tcd.ie</string>
    </map>
    <newObject name="robots-honoring-policy"
class="org.archive.crawler.datamodel.RobotsHonoringPolicy">
      <string name="type">classic</string>
      <boolean name="masquerade">false</boolean>
      <text name="custom-robots" />
      <stringList name="user-agents" />
    </newObject>
    <newObject name="frontier"
class="org.archive.crawler.frontier.BdbFrontier">
      <float name="delay-factor">5.0</float>
     <integer name="max-delay-ms">5000</integer>
     <integer name="min-delay-ms">500</integer>
     <integer name="max-retries">30</integer>
     <long name="retry-delay-seconds">900</long>
     <integer name="preference-embed-hops">1</integer>
     <integer name="total-bandwidth-usage-KB-sec">0</integer>
     <integer name="max-per-host-bandwidth-usage-KB-sec">0</integer>
     <boolean name="ip-politeness">false</boolean>
     <string name="force-queue-assignment" />
     <boolean name="pause-at-finish">false</boolean>
     <boolean name="hold-queues">true</boolean>
     <integer name="balance-replenish-amount">3000</integer>
     <long name="queue-total-budget">-1</long>
     <string name="cost-
policy">org.archive.crawler.frontier.ZeroCostAssignmentPolicy</string>
   </newObject>
   <map name="uri-canonicalization-rules">
     <newObject name="Lowercase"
class="org.archive.crawler.url.canonicalize.LowercaseRule">
       <boolean name="enabled">true</boolean>
     </newObject>
     <newObject name="Userinfo"
class="org.archive.crawler.url.canonicalize.StripUserinfoRule">
       <boolean name="enabled">true</boolean>
     </newObject>
     <newObject name="WWW"
class="org.archive.crawler.url.canonicalize.StripWWWRule">
       <boolean name="enabled">true</boolean>
     </newObject>
     <newObject name="SessionIDs"
class="org.archive.crawler.url.canonicalize.StripSessionIDs">
       <boolean name="enabled">true</boolean>
     </newObject>
     <newObject name="QueryStrPrefix"
class="org.archive.crawler.url.canonicalize.FixupQueryStr">
       <boolean name="enabled">true</boolean>
     </newObject>
   </map>
   <map name="pre-fetch-processors">
     <newObject name="Preselector"
class="org.archive.crawler.prefetch.Preselector">
       <boolean name="enabled">true</boolean>
       <map name="filters" />
```

```
      <boolean name="recheck-scope">true</boolean>
      <boolean name="block-all">false</boolean>
      <string name="block-by-regexp" />
    </newObject>
    <newObject name="Preprocessor"
class="org.archive.crawler.prefetch.PreconditionEnforcer">
      <boolean name="enabled">true</boolean>
      <map name="filters" />
      <integer name="ip-validity-duration-seconds">21600</integer>
      <integer name="robot-validity-duration-seconds">86400</integer>
    </newObject>
  </map>
  <map name="fetch-processors">
    <newObject name="DNS" class="org.archive.crawler.fetcher.FetchDNS">
      <boolean name="enabled">true</boolean>
      <map name="filters" />
      <boolean name="accept-non-dns-resolves">false</boolean>
    </newObject>
    <newObject name="HTTP" class="org.archive.crawler.fetcher.FetchHTTP">
      <boolean name="enabled">true</boolean>
     <map name="filters" />
     <map name="midfetch-filters" />
     <integer name="timeout-seconds">1200</integer>
     <integer name="sotimeout-ms">20000</integer>
     <long name="max-length-bytes">0</long>
     <string name="load-cookies-from-file" />
     <string name="save-cookies-to-file" />
     <string name="trust-level">open</string>
     <stringList name="accept-headers" />
     <string name="http-proxy-host">134.226.32.57</string>
     <string name="http-proxy-port">8080</string>
     <string name="default-encoding">ISO-8859-1</string>
     <boolean name="sha1-content">true</boolean>
     <boolean name="send-connection-close">true</boolean>
     <boolean name="send-referer">true</boolean>
     <boolean name="send-range">false</boolean>
   </newObject>
 </map>
 <map name="extract-processors">
   <newObject name="TextCatProcessor"
class="org.metacombine.languagemodule.TextCatProcessor">
     <boolean name="enabled">true</boolean>
     <map name="filters" />
   </newObject>
   <newObject name="BowProcessor"
class="org.metacombine.crawlmodule.BowProcessor">
     <boolean name="enabled">true</boolean>
     <map name="filters" />
   </newObject>
   <newObject name="ExtractorHTTP"
class="org.archive.crawler.extractor.ExtractorHTTP">
     <boolean name="enabled">true</boolean>
     <map name="filters" />
   </newObject>
   <newObject name="ExtractorHTML"
class="org.archive.crawler.extractor.ExtractorHTML">
     <boolean name="enabled">true</boolean>
     <map name="filters" />
   </newObject>
   <newObject name="ExtractorCSS"
class="org.archive.crawler.extractor.ExtractorCSS">
```

```
      <boolean name="enabled">true</boolean>
      <map name="filters" />
    </newObject>
    <newObject name="ExtractorJS"
class="org.archive.crawler.extractor.ExtractorJS">
      <boolean name="enabled">true</boolean>
      <map name="filters" />
    </newObject>
    <newObject name="ExtractorSWF"
class="org.archive.crawler.extractor.ExtractorSWF">
      <boolean name="enabled">true</boolean>
      <map name="filters" />
    </newObject>
  </map>
  <map name="write-processors">
    <newObject name="Archiver"
class="org.archive.crawler.writer.ARCWriterProcessor">
      <boolean name="enabled">true</boolean>
      <map name="filters" />
      <boolean name="compress">true</boolean>
      <string name="prefix">IAH</string>
      <string name="suffix">${HOSTNAME}</string>
      <integer name="max-size-bytes">70000000</integer>
      <stringList name="path">
        <string>arcs</string>
      </stringList>
      <integer name="pool-max-active">5</integer>
      <integer name="pool-max-wait">300000</integer>
      <long name="total-bytes-to-write">0</long>
    </newObject>
  </map>
  <map name="post-processors">
    <newObject name="Updater"
class="org.archive.crawler.postprocessor.CrawlStateUpdater">
      <boolean name="enabled">true</boolean>
      <map name="filters" />
    </newObject>
    <newObject name="Postselector"
class="org.archive.crawler.postprocessor.Postselector">
      <boolean name="enabled">true</boolean>
      <map name="filters" />
      <boolean name="seed-redirects-new-seed">true</boolean>
      <boolean name="override-logger">false</boolean>
      <map name="scope-rejected-uri-log-filters" />
    </newObject>
  </map>
  <map name="loggers">
    <newObject name="crawl-statistics"
class="org.archive.crawler.admin.StatisticsTracker">
      <integer name="interval-seconds">20</integer>
    </newObject>
  </map>
  <string name="recover-path" />
  <boolean name="recover-retain-failures">false</boolean>
  <newObject name="credential-store"
class="org.archive.crawler.datamodel.CredentialStore">
    <map name="credentials" />
  </newObject>
  </controller>
</crawl-order>
```

Appendix D – U-CREATe User Manual

U-CREATe

User-driven Content Retrieval, Exploration and Assembly Toolkit for eLearning

User Manual

Chapter 1. Creating and Editing Mind maps

This manual explains the user interface of U-CREATe. It provides information on creating, editing and formatting Mind maps, and nodes within Mind maps.

Exploring the U-CREATe Interface

It is essential for the user to be familiar with the U-CREATe user interface and the basics of U-CREATe. The user interface has three main components:

- The Menu bar
- The Formatting bar
- The Icon toolbar

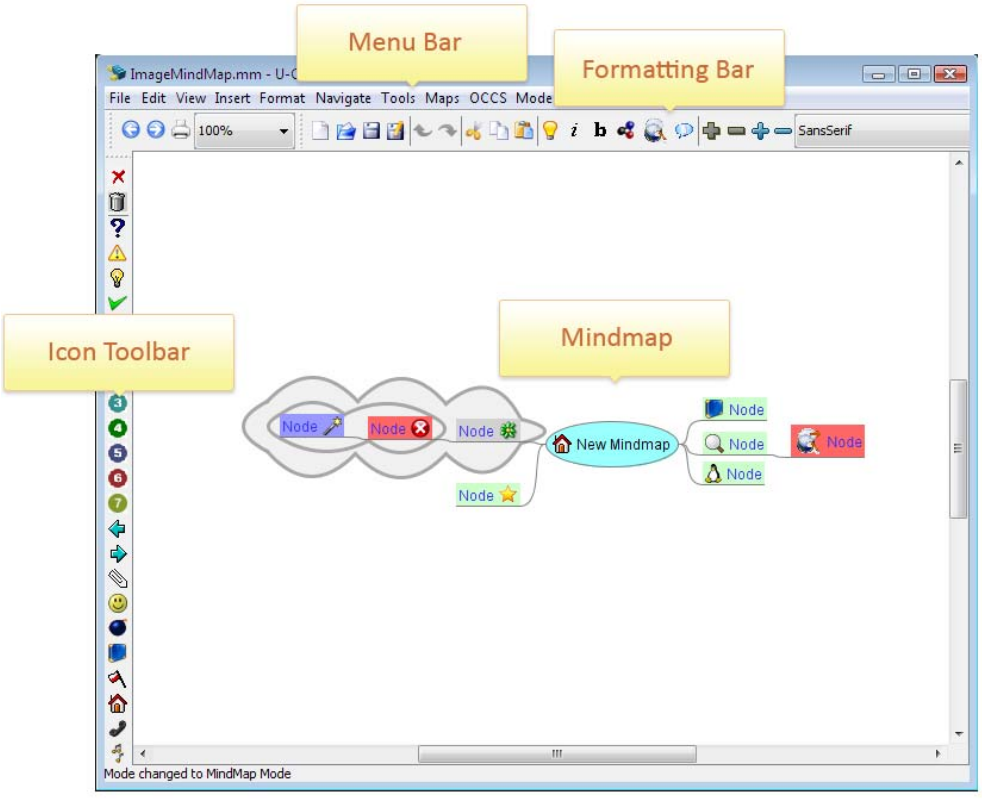


Figure 1-1. U-CREATe Interface

Before creating a new Mind map, the user should be aware of some terminology which will be used both in the system and throughout this document. A node is a graphical text box that is used to store information. Nodes are connected together using lines called edges. When a new Mind map is created in U-CREATe, a grey, oval shaped node with the label “New Mind map” appears in the centre of the interface window. This is the root node. Mind maps are constructed by adding nodes to the root node. The root node is selected when it is highlighted in grey. Child nodes originate from the parent node and are positioned one level lower. Sibling nodes are positioned at the same level as the highlighted node. However, a sibling node may not be created for the root node. The root node can only have child nodes. Any other node in the Mind map can have parent, child and sibling nodes.

As a Mind map is built, nodes and edges can be emphasized through the addition of colours, sizes, fonts and other attributes. “Clouds” can be used to group together interrelated nodes, icons can also be added to a node. These features all help to customise Mind maps. Nodes can be "folded" or "unfolded" by clicking on them. Unfolding a node displays all the generations of nodes below it.

Creating and Deleting Nodes

The first step in creating a new Mind map is to create nodes. The process of node creation follows the following general steps:

- Click File on the menu bar and select New. A new screen appears, with a root node “New Mind map”.
- Right-click on the root node and select New Child Node. Alternatively click Insert on the menu bar and select New Child Node. U-CREATe opens an edit box which allows the user to name the new child node. The new node is inserted under the root node.
- Right click on this node and select New Sibling Node. Again an edit box is opened. Once the node is named, a sibling node is placed at the same level as the selected node.

When generating a Mind map it will be necessary on occasion to delete nodes. This be achieved by placing the mouse pointer over the node and pressing the delete key on the keyboard. Alternatively, it is possible to right-click on the node and select Remove Node.

Right-clicking a node and using the node context menu can be used to perform the majority of editing, inserting, and formatting operations.

Saving a Mind map

A Mind map may be saved via the File menu or through the Save Icon on the Formatting Bar. The first time a Mind map is saved the user is asked to select the location in which to place the file. Save As can be used if the user wishes to subsequently save the Mind map in an alternative location. By default, U-CREATe Mind maps will have the file extension .mm.

Editing Node Text

Many operations can be performed which edit the text in a node. The following is a set of commonly used editing operations:

- Highlight a node with the mouse pointer and press F2, the End key or the Home key. This opens an Edit Node box which allows the text to be edited.
- Right-click on the root node and select Edit Node. This opens an Edit Node box which allows the text to be edited.
- Click Edit on the menu bar and select Edit Node. This opens an Edit Node box which allows the text to be edited.

In the Edit Node box, it is possible to split a node in two. Simply select the text that you want to appear in one node and press split. This places the selected text in a new node and the remaining text in the existing node.

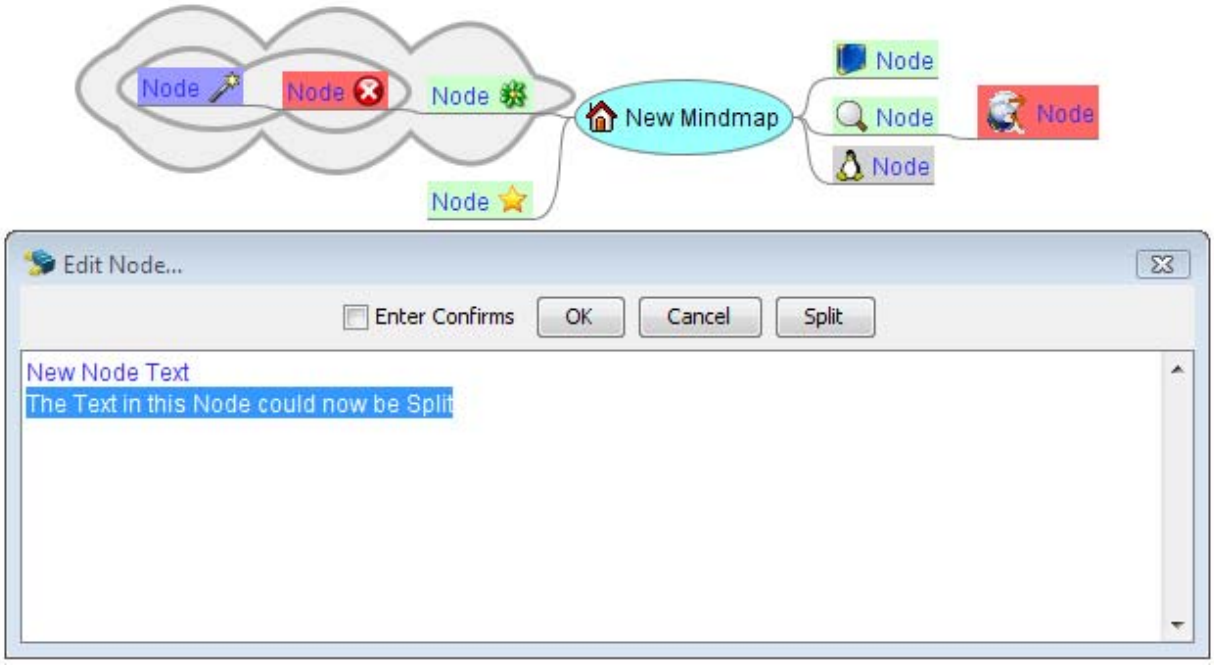


Figure 1-2. Node Editing

To insert a special symbols into a node in U-CREATe the user must first generate the symbol in MS Word or any other text editor. The symbol can then be copied from the text editor to U-CREATe. U-CREATe fully supports Unicode, a standard for identifying letters and numbers that attempts to include character sets from all languages around the world. Thus, you can use the script of your choice.

Formatting a Node

Formatting a node is the process of modifying the colour, shape, size, font and other attributes of the node. This aids the user in distinguishing between various types of nodes within a Mind map.

Styles can be applied to them nodes. Currently there are two styles available in U-CREATe as standard, Fork and Bubble. The styles are merely different visual representations of the node, they do not affect functionality. The Bubble style uses an oval shaped bubble to enclose the data of a node, whereas the Fork style underlines the data without enclosing it. Styles can be selected by right-clicking on a node and selecting the Format Menu, or by clicking Format on the menu bar, and selecting the required style.

The text within a node can also be formatted. Font size, font colour and font family can all be adjusted. A style, such as bold or italic can also be applied to the font. Size, family, and style can all be modified via the formatting bar. There are drop down lists for font family and font size. There are also bold and italic icons. These attributes can also be edited via the Format menu or via right-click and selecting format, where it is also possible to modify text colour using the Node Colour option.

In addition to altering the colour of the text within a node, it is also possible to change the background colour of the node itself. By selecting a node and clicking Format on the menu bar, or by right-clicking a node and selecting Format, the user is presented with the Node Background Colour option. This presents a palette of colours from which the user can select the node colour.

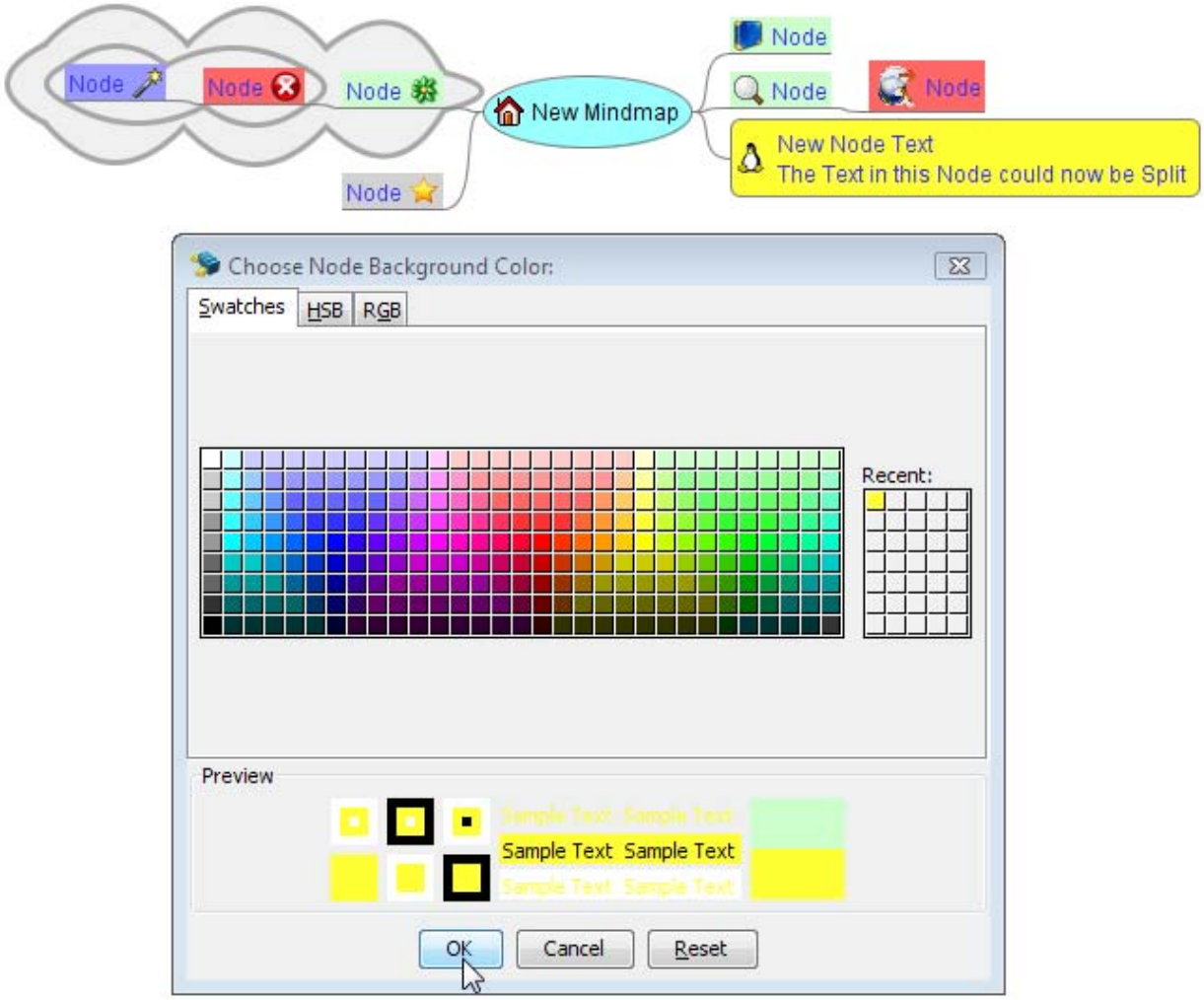


Figure 1-3. Node Background Colour Formatting

There is an Automatic Layout option within U-CREATe which fixes the layout of a Mind map. It assigns pre-defined colours to represent the different stages of the map. The default colours used are black for the first level of nodes and blue for the second level of the Mind map. This option is available in the Format menu, or by right-clicking on a node and selecting Format.

Formatting Edges

Edges can also be formatted to alter their colour, style and width. All of these options are available through the Format menu on the menu bar. Colour can be selected from a palette similar to node colour and background colour. There are four styles currently available in U-CREATe; Linear, Bezier, Sharp Linear, Sharp Bezier. The linear style refers to straight interconnecting lines between nodes. Bezier edges are curved on approach to each node and

tend to be more visually appealing. The sharp option of both styles narrows the width of the edge as it moves from parent to child. This creates the appearance of direction in the edge.

Using Physical Styles

Physical styles are referred to as “Patterns” in U-CREATe. The standard patterns are preset font, node and edge variations encompassing style, size and colour. However, patterns can be designed which include any formatting option, such as the addition of icons or links to certain images or files. To apply a physical style click on Format on the menu bar and select Physical Style.

The file “patterns.xml” located in the “U-CREATe” folder of your home directory contains the code for every physical style. A user can add physical styles by editing the code in this file. If the code has a <node> tag, then the pattern applies to a node, similarly if the code has an <edge> tag, then the pattern applies to the edge. A typical pattern in patterns.xml looks like this:

```
<pattern name="Folder">
<node colour="#CC9900">
<font name="Arial" size="14" />
</node>
</pattern>
```

Note that the <font> tag is a child of <node>. The <font> tag is terminated by the "/". It is possible to have simple patterns that change various attributed of a node. A tag can include imbedded links. In this case it is necessary to use the "&" codes for special HTML characters. The following example attaches a graphic to the node that has been stored in the U-CREATe images directory.

```
<pattern name="Question mark">
<node TEXT="&lt;html&gt;&lt;imgsrc=&quot;file://
C:/ProgramFiles/U-CREATe/Images/question.gif&quot;&gt
;&lt;/html&gt; ">
<font name="Default" size="14"/>
</node></pattern>
```

Be aware that the use of the "text" attribute in a pattern, whether to insert actual text or HTML code, will completely overwrite any existing text in that node. It is also important to note that the shortcut keys for physical styles are assigned according to their sequence in the

patterns.xml list. Therefore any additions or deletions in the file can cause the numbering to change.

Searching Nodes

The Find option in U-CREATe searches a node and its descendants. The search is performed in a breadth-first manner. By default, a search is not conducted across an entire Mind map. It is only executed on the selected node and its descendants. A node is selected by placing the mouse pointer over the node, then to conduct a search press Control+F, or alternatively choose Find from the Edit menu. This opens a search dialogue box, where the search terms can be entered. U-CREATe selects the first occurrence of a node containing the desired text. To find the next match simply press Control+G or select Find Next from the Edit Menu. It is possible to search the whole Mind map by selecting the root node before performing the search.

Chapter 2. Adding Clouds, Links and Icons

Maps can have groups of nodes highlighted through the addition of a graphical feature called a cloud. An additional knowledge layer can be added to U-CREATe Mind maps through the addition of hyperlinks to OCCS topic-specific educational content, web sites or files in your local directory. Nodes can be labelled and categorised using icons. Graphical links can also be applied to link nodes which may not be in the same hierarchical branch.

Highlighting Nodes with Clouds

Clouds are well suited for highlighting specific regions of a Mind map. When a cloud is applied to a node, the node and all its descendants are encompassed by the cloud. Clouds can have different background colours. Highlighting nodes through the use of clouds can help to categorise the nodes within a map, grouping together nodes with common attributes or features. To add a cloud to a node, select and node in question, click Insert on the menu bar and select Cloud. Alternately right-click on the node, select Insert, and click Cloud. U-CREATe will then insert a cloud for the selected node and all its decendants.

Browsing the OCCS

As a Mind map is under construction, the user can browse the OCCS to find educational content related to the subject area or concept they are mapping. This aids the user in generating their map and can add extra, in-depth and detailed information to a node. To browse the OCCS while generating a map, the user can click on the OCCS icon on the

formatting bar, or alternately right-click and select Browse OCCS. U-CREATe then opens the OCCS search interface in the users default web browser.

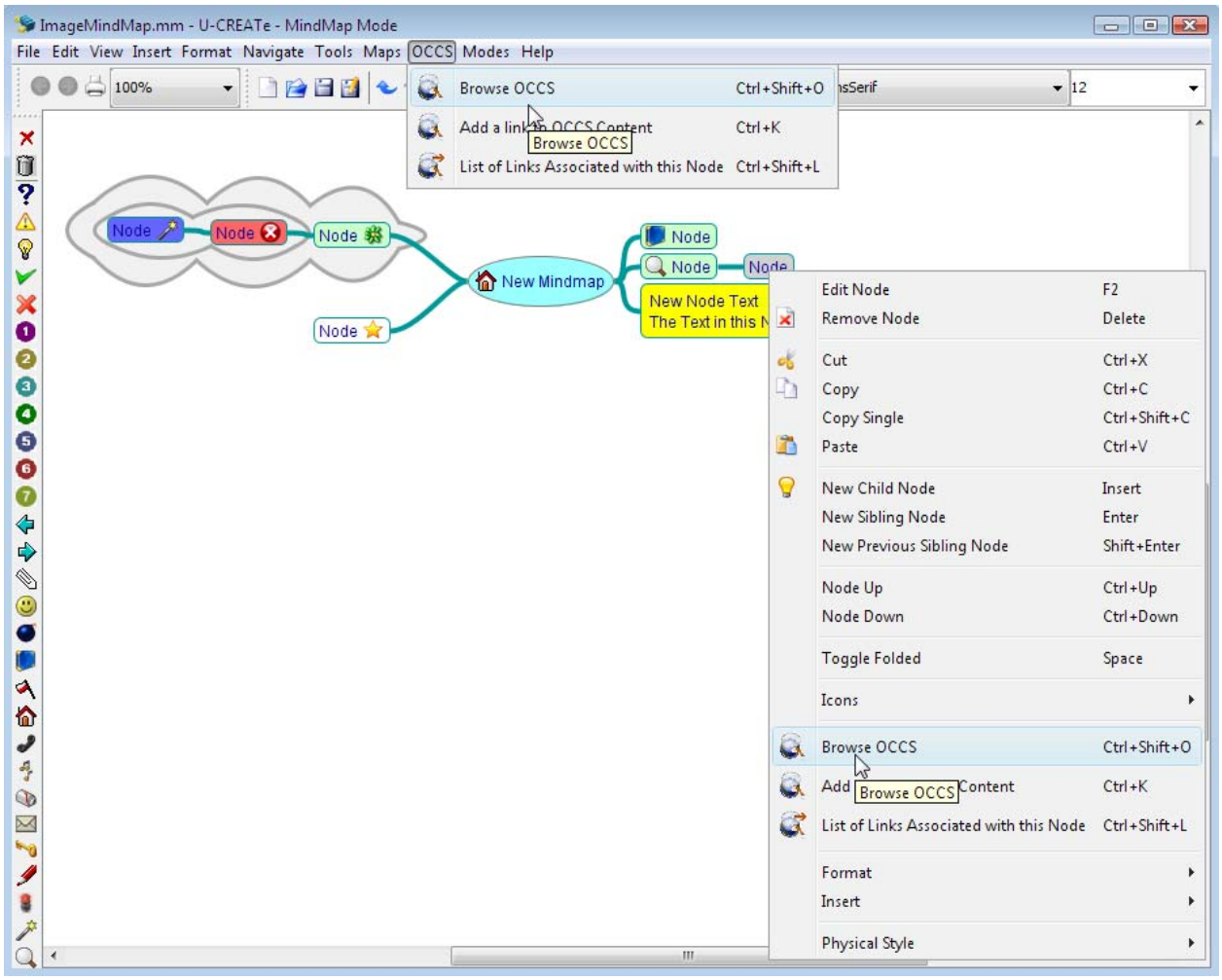


Figure 2-1. Browsing the OCCS

Adding Links to a Node

Links can be added to a node which point to OCCS content, web pages, local files, or e-mail addresses. There are many ways in which a link may be added to a node:

- Right-click on a node and select Add a Link to OCCS Content
- Right-click on a node, select Insert and then click Hyperlink (File Chooser)
- Right-click on a node, select Insert and then click Add Local Hyperlink
- Click OCCS on the menu bar and select Add a Link to OCCS Content
- Click Insert on the menu bar and select Hyperlink (File Chooser)
- Click Insert on the menu bar and select Add Local Hyperlink

U-CREATe then displays a text input box where a URL or file location can be pasted or typed. In the case of Hyperlink (File Chooser) and explorer window is opened to enable the user to browse for the required file in the local file system. Once a link is added U-CREATe adds an icon to the node depending on what type of link is added. For instance an OCCS icon is added when OCCS links are present in a node and an envelope icon is added when email links are included in a node.

When a node containing a hyperlink is selected, the mouse pointer changes to a hand symbol over the link icon. When the node is clicked, an Associated Content Hyperlinks box is displayed. This box lists every link associated with that node. Nodes can be removed by clicking the delete icon located next to each link.

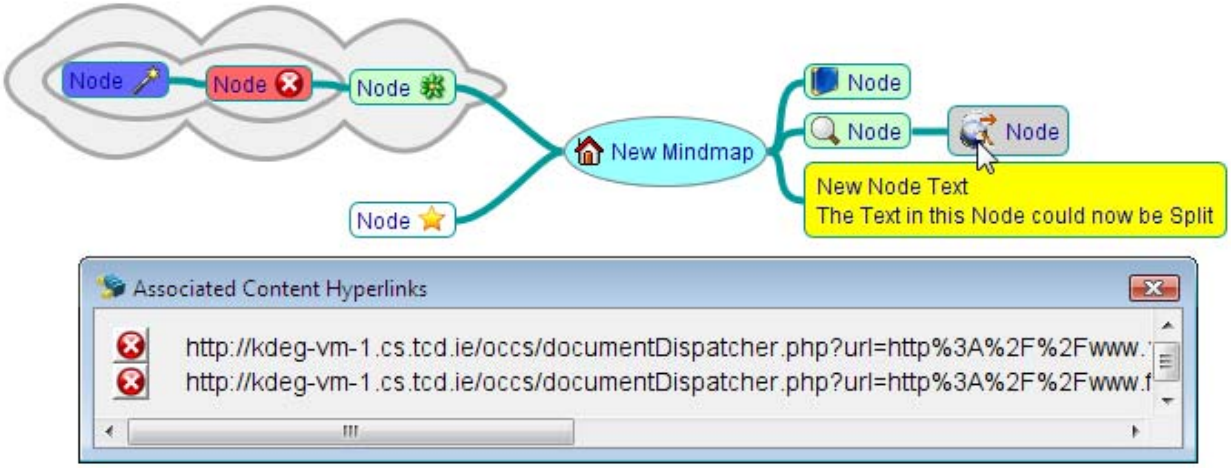


Figure 2-2. Nodes Associated Content Links

Adding Icons to a Node

Icons can be added to nodes to indicate particular attributes or tasks related to that node. They may also be used to distinguish between categories or classes of nodes. Icons can be added to a node through the icon bar, or alternately through the Select Icon option on the formatting bar. Each node may have many associated icons. U-CREATe offers 35 individual icons for inclusion in a Mind map. The icon toolbar can be hidden from view if necessary by clicking View on the menu bar and selecting Toggle Left Toolbar. When an icon is selected, it is inserted in the node in order of addition. To remove an icon from a node, select the node in question using the mouse pointer and click the red cross at the top of the icon bar. This will remove the last icon that was inserted. Alternately right-click on the node, select Icons and click Remove Last Icon or Remove All Icons.

Adding Graphical Links

A graphical link is a mono or bi-directional path between nodes in a Mind map. Graphical links can be created between two or more nodes. To add a graphical link, select two or more nodes by holding the Control key and clicking on the nodes in question. Once the required node are selected, click Insert on the menu bar or right-click and select Insert, then click Add Graphical Link. U-CREATe inserts a graphical link between the nodes.

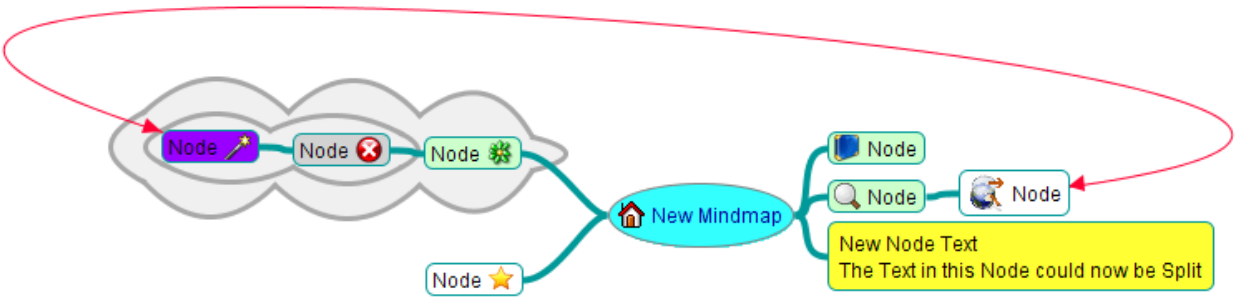


Figure 2-3. Graphical Links

The colour and direction of a graphical link can be altered by right-clicking on the link. Graphical links can be deleted by right-clicking on the link and selecting Remove Arrow Link. The route a graphical link takes can be modified by clicking on the arrow and dragging it around the map to find the required path.

Chapter 3. Advanced Operations

U-CREATe provides advanced functionality for the experienced user. This includes exporting Mind maps in various formats for use in other systems. It is possible to customise U-CREATe, and use advanced features such as rich text and images in nodes. U-CREATe provides some features which can help to minimize a users workload.

Selecting Multiple Nodes

Some tasks within U-CREATe require the selection of multiple nodes in a map. To select particular nodes, simply hold the Control key and click on the nodes in question. It is also possible to select a complete sub-tree of a map. To select a sub-tree, hold the Alt key and click the parent node of the sub-tree. Alternately hold the Shift key while moving through the sub-tree with the keyboard arrow keys. To cancel a selection of multiple nodes, simply click the map background. To select all the visible nodes of a Mind map, select Edit on the menu bar and click Select All.

Dragging and Dropping

Nodes can be moved around within a Mind map using drag and drop functionality. Nodes can be dropped as either siblings or children. To drop a node as a child, click on the node in question and drag it to the destination node. Drop the node on the outer side of the destination node. U-CREATe then adds the selected node, and all its descendants, as children of the destination node.

To drop a node as a sibling, drop the node on the inner side of the destination node.

Copy a Node During Drag and Drop

Nodes may be copied rather than moved by drag and drop. To copy a node, and its descendants, hold the Control key while dragging. U-CREATe copies the selection to the new location.

Creating a Graphical Link by Drag and Drop

To create a graphical link using drag and drop, select the first node. While holding the right mouse button, drag and drop the node onto the target node. U-CREATe will create a graphical link from the first node to the target node.

Copying and Pasting

Multiple nodes can be copied and pasted in Mind maps. Nodes can be copied with or without its descendants. Copy and Copy Single are options in the Edit menu for this functionality in U-CREATe. To copy a node with its descendants, select Edit on the menu bar and click Copy. U-CREATe will copy the node in question and all its sub-nodes. To copy a node without its descendants, click the Copy Single option.

Normal text or HTML can be copied and pasted or dropped directly into nodes from external applications. This data can include pieces of text from Internet Explorer or files from the Windows operating system. Plain text, HTML, and file lists can be pasted from Windows Explorer into U-CREATe. It should be noted that when multiple lines of plain text are pasted into U-CREATe, they are entered as multiple nodes. The new nodes are entered as child nodes of the currently selected node. When HTML is pasted into U-CREATe, it is pasted as plain text. The links in HTML are pasted as children of an additional node with text “Links”.

When a Mind map branch is copied from U-CREATe and pasted into a RTF editor, the associated formatting, including colour and font, is also preserved. Hyperlinks are pasted in

angle brackets (< >). RTF editors include Microsoft Word, WordPad, Microsoft Outlook, and some tabbed Linux notebooks.

Folding and Unfolding

Nodes can have their descendants or sub-nodes folded and hidden from view. Upon folding, the parent node remains intact, but the descendants are temporarily hidden. This concept is useful when dealing with large Mind maps containing large volumes of information. A folded node is marked with a small circle on its outer edge. To fold the descendants of a node, move the mouse pointer over the parent node and press Space. Alternately, right click on the node and select Toggle Folded. U-CREATe automatically folds the descendants of the selected node from view. To unfold a node's descendants simply use the Space key or the Toggle Folded option again.

When a Mind map has multiple levels of nodes, it is possible to fold and unfold them level-by-level. To fold the descendants of a node one level at a time, select the node and press Alt+PageUp. To unfold the node one level at a time, select the node and press Alt+PageDown. To fold all the nodes in a Mind map, select the root node and click the grey minus sign on the formatting bar. To unfold all the nodes in the map, select the root node, and click the grey plus sign on the formatting bar.

Using Undo

The Undo feature can be used to cancel the result of a previous action. To set the number of actions that are stored for undoing, select Tools on the menu bar and click Preferences. In the U-CREATe properties window, click Behavior and select Undo Levels. To undo the previous completed action, click Edit on the menu bar and select Undo. To redo the previous action, select Redo.

Navigation Features

There are different techniques employed to navigate through a Mind map in U-CREATe.

Moving Around a Map

There are multiple methods of moving around a Mind map. The keyboard arrow keys or the Page Up and Page Down keys can be used. Clicking the map background and dragging the map around is also useful for navigating a map. To move to the top of the current sub-tree, press Page Up. To move to the bottom of the current sub-tree, press Page Down. To move to

the root node, press Escape. To position a node freely, drag the node by the invisible handle placed on the inside of the node.

Alternating Between Open Mind maps

U-CREATe can be used to edit numerous Mind maps simultaneously. It is possible to switch between the open maps. Fast switching among maps can be completed by any of the following methods:

- Right-click on an empty area of the map and select the map name.
- Click Maps on the menu bar and choose Previous Map or Next map.
- Press Control+Left Arrow or Control+Right Arrow.
- Click Maps on the menu bar and choose the map name.

Scrolling the map

A Mind map may be scrolled using any of the following methods:

- Using the standard window scroll bars.
- Clicking on the map background and dragging it in the desired direction.
- Using the mouse wheel. To scroll horizontally, hold the Shift key or one of the mouse buttons and move the mouse wheel.

Zooming

It is possible to zoom in or out of a Mind map view by any of the following methods:

- Using the zoom icons on the formatting toolbar.
- Using the mouse wheel while holding the Control key.
- Press Alt+Up Arrow or Alt+Down Arrow.

Exporting and Importing Data

U-CREATe gives you the option to export and import data. If you want to view a Mind map in a different format such as HTML, XHTML, or any picture format, you can use the export feature available in U-CREATe.

Exporting to HTML or XHTML

It is possible to export an entire Mind map or individual branches of a Mind map to HTML. To export a branch as HTML, press Control+H. To export an entire map as HTML, click File on the menu bar and select Export. Choose the option As HTML. U-CREATe exports the Mind map to HTML and allows the user to select the location to save the resulting file.

XHTML is a general-purpose mark-up language that utilises XML elements to present information on the WWW. XHTML focuses on structuring documents rather than presenting them, thus allowing the information to be presented in a variety of different forms, based on user need and device capability. To export a map as XHTML (JavaScript version), click File on the menu bar and select Export. Choose the option As XHTML (JavaScript version). The user can select the location to save the resulting file.

Exporting to PNG or JPEG

A Mind map can be exported from U-CREATe as a PNG or JPEG image. PNG (Portable Network Graphics) is a bitmapped graphics file format endorsed by the World Wide Web Consortium (W3C). PNG provides advanced graphics features such as 48-bit colour, built-in colour correction, tight compression, and the ability to display at one resolution and print at another. To export the map as a PNG image, click File on the menu bar and select Export. Choose the option As PNG. The user can then select the desired location in which to save the PNG image. JPEG (Joint Photographic Experts Group) is a compressed graphic file normally used for images that require many colours. To export a map as a JPEG image, click File on the menu bar and select Export. Choose the option As JPEG.

Exporting to Open Office

To export a map to an open office 1.4 writer document, click File on the menu bar and select Export. Choose the option As Open Office Writer Document. The user can select the desired location to save the resulting open office document.

Importing Data

Folder trees and Internet Explorer favourites can be imported into U-CREATe. To importing a folder structure, click File on the menu bar and select Import. Choose the option Folder Structure. U-CREATe displays a dialogue box stating “Select the folder to import”. Choose the required folder and click Open. U-CREATe imports each folder in the tree as a node with a link to the folder location on the local drive.

To importing Internet Explorer favourites, click on File on the menu bar and select Import. Choose the option Explorer Favourites. U-CREATe displays a dialogue box stating, “Select the folder, in which your favourites reside”. Select the required folder. U-CREATe imports the favourites to the selected node.

Integrating with Word

It is possible to copy an entire Mind map or an individual branch of a map to another application that can understand a rich text format (RTF). When a map is transferred, text formatting and links associated with nodes are preserved. U-CREATe maps can be copied into Microsoft Word, WordPad, or Outlook messages. There are various methods of integrating a Mind map with MS Word, the most efficient method is to export the map as HTML and copy the HTML content into an MS Word document. When pasting into MS Word, select Edit and click Paste Special. Word displays a dialogue box. Choose the option HTML format and click OK.

Printing

A Mind map can be printed, shrinking the entire map to fit on one page, or across several sheets. Landscape page orientation option makes more efficient use of space compared to portrait. A print preview option is available to view the print layout before sending a map to a printer. If a Postscript printer or generic Postscript driver is available, a Mind map can be printed to a file. The Postscript file can be viewed with Ghost view or similar software.

If a Mind map has been exported to HTML, it can subsequently be printed from an internet browser. An exported map can also be printed from MS Word or WordPad. To print a Mind map click File on the menu bar and select Print. Alternately click Page Setup and U-CREATe will display a Print Scaling dialogue box. Select the option Fit to One Page or enter a Print Zoom Factor. The Page Setup dialogue box will then open. In the Orientation section, select Landscape. Adjust any other information as desired. Click on File and select Print.

Setting Preferences

U-CREATe can be customised by the setting of user preferences. These preferences can include language, keyboard mappings, default font style, appearance, behaviour when exporting to HTML and many more categories. These changes become visible upon U-CREATe re-start. To modify user preferences click Tools on the menu bar and select Preferences. This opens the U-CREATe Properties dialogue box opens. Click the information categories on the left side, make the required changes and click Save.

Rich Text in Nodes

U-CREATe allows the addition of HTML to the nodes of a Mind map. Nodes starting with <html> are rendered using the HTML contained in them. HTML can be useful for the

inclusion of lists, tables or formatted text in nodes. It should be noted that there is no support for the export of HTML or pictures contained in nodes .

Images in Nodes

U-CREATe supports the addition of images in nodes. U-CREATe supports PNG, JPEG, and GIF image formats. When an image is inserted into a node which already contains text, that text is overwritten and lost. Images inserted in this way are not correctly pasted outside U-CREATe, and cannot be accurately exported to HTML.

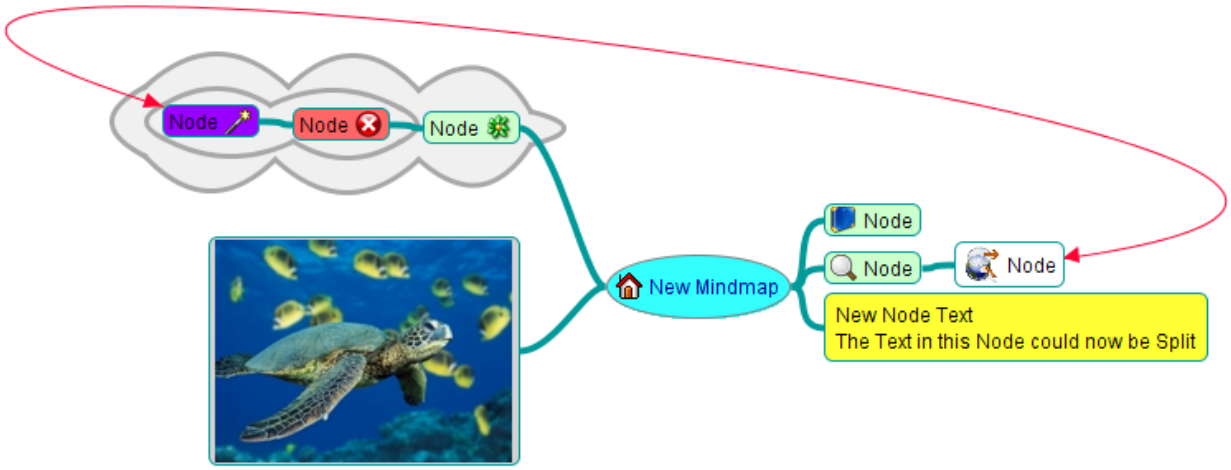


Figure 3-1. Graphical Links

To insert an image into a node, click Insert on the menu bar and select Image (File Chooser or Link). Alternately right-click a node, select Insert and click on Image (File Chooser or Link). U-CREATe displays the Open dialogue box to allow the user to specify the location of the image.

HTML can also be used to insert images into nodes. If the map and the image are located in the same directory, it is sufficient to specify the image name, for example <html><img src=”image.jpg”></html>. If the Mind map and the image are located in different directories the exact location of the file must be specified, for example <html><img src=”c:/folder/user/image.jpg”></html>.

Using Experimental File Locking

Experimental file locking ensures multiple users do not edit the same Mind map simultaneously. This prevents the accidental overwriting information. Experimental file locking is disabled by default. To enable experimental file locking click Tools on the menu bar and select Preferences. In the U-CREATe Properties dialogue box click Environment. In

the Files section, enable the option Experimental File Locking and save. A reboot must be completed for the changes to take effect.

Keyboard Shortcuts

The Majority of functions in U-CREATe can be performed using keyboard shortcuts. The following is a complete listing of all the keyboard shortcuts currently implemented.

New map Ctrl+N

Open map Ctrl+O

Save map Ctrl+S

Save as Ctrl+A

Print Ctrl+P

Close Ctrl+W

Quit Ctrl+Q

Previous map Ctrl+LEFT

Next map Ctrl+RIGHT

Export file to HTML Ctrl+E

Export branch to HTML Ctrl+H

Export branch to new MM file ALT+A

Open first file in history Ctrl+Shift+W

Edit commands

Find Ctrl+F

Fine Next Ctrl+G

Cut Ctrl+X

Copy Ctrl+C

Copy Single Ctrl+Y

Paste Ctrl+V

Mode commands

Mind map mode Alt+1

Browse mode Alt +2

File mode Alt+3

Node formatting commands

Italicize Ctrl+I

Bold Ctrl+B

Cloud Ctrl+Shift+B

Change node color Alt+C

Blend node color Alt+B

Change node edge color Alt+E

Increase node font size Ctrl+L

Decrease node font size Ctrl+M

Increase branch font size Ctrl+Shift+L

Decrease branch font size Ctrl+Shift+M

Node navigation commands

Go to root ESCAPE

Move up UP

Move down DOWN

Move left LEFT

Move right RIGHT

Follow link Ctrl+ENTER

Zoom out Alt+UP

Zoom in Alt+DOWN

New node commands

Add sibling node ENTER

Add child node INSERT

Add sibling before Shift+ENTER

Node editing commands

Edit selected node F2

Edit long node Alt+ENTER

Join nodes Ctrl+J

Toggle folded SPACE

Toggle children folded Ctrl+SPACE

Set link by file chooser CTRL+Shift+K

Add a link to OCCS Content CTRL+K

Set image by filechooser Alt+K

Move node up Ctrl+UP

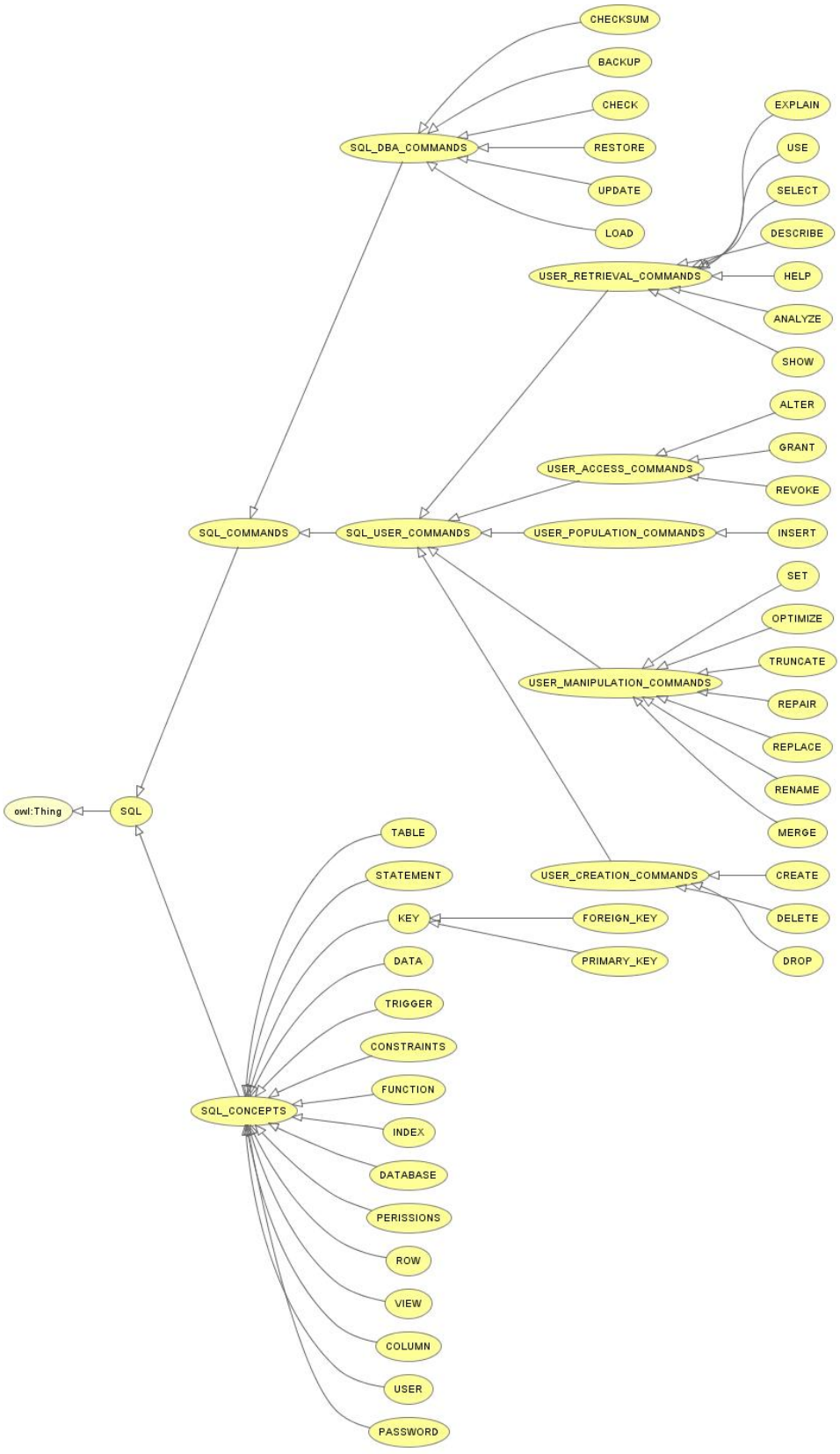
Move node down Ctrl+DOWN

Browse the OCCS Ctrl+Shift+O

View a Node’s Associated Hyperlinks Ctrl+Shift+L

Appendix E – OCCS Training Inputs and Outputs

Domain Ontology



Keywords

SQL
SQL ALTER
SQL REVOKE
SQL GRANT
SQL TRUNCATE
SQL OPTIMIZE
SQL REPAIR
SQL SET
SQL MERGE
SQL RENAME
SQL REPLACE
SQL DELETE
SQL DROP
SQL CREATE
SQL EXPLAIN
SQL SELECT
SQL SHOW
SQL USE
SQL DESCRIBE
SQL ANALYZE
SQL HELP
SQL INSERT
SQL RESTORE
SQL CECKSUM
SQL CHECK
SQL BACKUP
SQL LOAD
SQL UPDATE
SQL DATA
SQL CONSTRAINTS
SQL KEY
SQL FOREIGN_KEY
SQL PRIMARY_KEY
SQL TABLE
SQL COLUMN
SQL VIEW
SQL STATEMENT
SQL PERMISSION
SQL INDEX
SQL PASSWORD
SQL FUNCTION
SQL TRIGGER
SQL USER
SQL DATABASE
SQL ROW
SQL ALTER COMMAND
SQL REVOKE COMMAND
SQL GRANT COMMAND
SQL TRUNCATE COMMAND
SQL OPTIMIZE COMMAND
SQL REPAIR COMMAND
SQL SET COMMAND
SQL MERGE COMMAND
SQL RENAME COMMAND
SQL REPLACE COMMAND
SQL DELETE COMMAND
SQL DROP COMMAND
SQL CREATE COMMAND

SQL EXPLAIN COMMAND
SQL SELECT COMMAND
SQL SHOW COMMAND
SQL USE COMMAND
SQL DESCRIBE COMMAND
SQL ANALYZE COMMAND
SQL HELP COMMAND
SQL INSERT COMMAND
SQL RESTORE COMMAND
SQL CECKSUM COMMAND
SQL CHECK COMMAND
SQL BACKUP COMMAND
SQL LOAD COMMAND
SQL UPDATE COMMAND
SQL DATA COMMAND
SQL CONSTRAINTS COMMAND
SQL KEY COMMAND
SQL FOREIGN_KEY COMMAND
SQL PRIMARY_KEY COMMAND
SQL TABLE COMMAND
SQL COLUMN COMMAND
SQL VIEW COMMAND
SQL STATEMENT COMMAND
SQL PERMISSION COMMAND
SQL INDEX COMMAND
SQL PASSWORD COMMAND
SQL FUNCTION COMMAND
SQL TRIGGER COMMAND
SQL USER COMMAND
SQL DATABASE COMMAND
SQL ROW COMMAND
SQL ALTER CONCEPT
SQL REVOKE CONCEPT
SQL GRANT CONCEPT
SQL TRUNCATE CONCEPT
SQL OPTIMIZE CONCEPT
SQL REPAIR CONCEPT
SQL SET CONCEPT
SQL MERGE CONCEPT
SQL RENAME CONCEPT
SQL REPLACE CONCEPT
SQL DELETE CONCEPT
SQL DROP CONCEPT
SQL CREATE CONCEPT
SQL EXPLAIN CONCEPT
SQL SELECT CONCEPT
SQL SHOW CONCEPT
SQL USE CONCEPT
SQL DESCRIBE CONCEPT
SQL ANALYZE CONCEPT
SQL HELP CONCEPT
SQL INSERT CONCEPT
SQL RESTORE CONCEPT
SQL CECKSUM CONCEPT
SQL CHECK CONCEPT
SQL BACKUP CONCEPT
SQL LOAD CONCEPT
SQL UPDATE CONCEPT
SQL DATA CONCEPT
SQL CONSTRAINTS CONCEPT

SQL KEY CONCEPT
SQL FOREIGN_KEY CONCEPT
SQL PRIMARY_KEY CONCEPT
SQL TABLE CONCEPT
SQL COLUMN CONCEPT
SQL VIEW CONCEPT
SQL STATEMENT CONCEPT
SQL PERMISSION CONCEPT
SQL INDEX CONCEPT
SQL PASSWORD CONCEPT
SQL FUNCTION CONCEPT
SQL TRIGGER CONCEPT
SQL USER CONCEPT
SQL DATABASE CONCEPT
SQL ROW CONCEPT
SQL ALTER CONSTRUCT
SQL REVOKE CONSTRUCT
SQL GRANT CONSTRUCT
SQL TRUNCATE CONSTRUCT
SQL OPTIMIZE CONSTRUCT
SQL REPAIR CONSTRUCT
SQL SET CONSTRUCT
SQL MERGE CONSTRUCT
SQL RENAME CONSTRUCT
SQL REPLACE CONSTRUCT
SQL DELETE CONSTRUCT
SQL DROP CONSTRUCT
SQL CREATE CONSTRUCT
SQL EXPLAIN CONSTRUCT
SQL SELECT CONSTRUCT
SQL SHOW CONSTRUCT
SQL USE CONSTRUCT
SQL DESCRIBE CONSTRUCT
SQL ANALYZE CONSTRUCT
SQL HELP CONSTRUCT
SQL INSERT CONSTRUCT
SQL RESTORE CONSTRUCT
SQL CECKSUM CONSTRUCT
SQL CHECK CONSTRUCT
SQL BACKUP CONSTRUCT
SQL LOAD CONSTRUCT
SQL UPDATE CONSTRUCT
SQL DATA CONSTRUCT
SQL CONSTRAINTS CONSTRUCT
SQL KEY CONSTRUCT
SQL FOREIGN_KEY CONSTRUCT
SQL PRIMARY_KEY CONSTRUCT
SQL TABLE CONSTRUCT
SQL COLUMN CONSTRUCT
SQL VIEW CONSTRUCT
SQL STATEMENT CONSTRUCT
SQL PERMISSION CONSTRUCT
SQL INDEX CONSTRUCT
SQL PASSWORD CONSTRUCT
SQL FUNCTION CONSTRUCT
SQL TRIGGER CONSTRUCT
SQL USER CONSTRUCT
SQL DATABASE CONSTRUCT
SQL ROW CONSTRUCT
SQL ALTER DEFINITION

SQL REVOKE DEFINITION
SQL GRANT DEFINITION
SQL TRUNCATE DEFINITION
SQL OPTIMIZE DEFINITION
SQL REPAIR DEFINITION
SQL SET DEFINITION
SQL MERGE DEFINITION
SQL RENAME DEFINITION
SQL REPLACE DEFINITION
SQL DELETE DEFINITION
SQL DROP DEFINITION
SQL CREATE DEFINITION
SQL EXPLAIN DEFINITION
SQL SELECT DEFINITION
SQL SHOW DEFINITION
SQL USE DEFINITION
SQL DESCRIBE DEFINITION
SQL ANALYZE DEFINITION
SQL HELP DEFINITION
SQL INSERT DEFINITION
SQL RESTORE DEFINITION
SQL CECKSUM DEFINITION
SQL CHECK DEFINITION
SQL BACKUP DEFINITION
SQL LOAD DEFINITION
SQL UPDATE DEFINITION
SQL DATA DEFINITION
SQL CONSTRAINTS DEFINITION
SQL KEY DEFINITION
SQL FOREIGN_KEY DEFINITION
SQL PRIMARY_KEY DEFINITION
SQL TABLE DEFINITION
SQL COLUMN DEFINITION
SQL VIEW DEFINITION
SQL STATEMENT DEFINITION
SQL PERMISSION DEFINITION
SQL INDEX DEFINITION
SQL PASSWORD DEFINITION
SQL FUNCTION DEFINITION
SQL TRIGGER DEFINITION
SQL USER DEFINITION
SQL DATABASE DEFINITION
SQL ROW DEFINITION
SQL ALTER EXAMPLE
SQL REVOKE EXAMPLE
SQL GRANT EXAMPLE
SQL TRUNCATE EXAMPLE
SQL OPTIMIZE EXAMPLE
SQL REPAIR EXAMPLE
SQL SET EXAMPLE
SQL MERGE EXAMPLE
SQL RENAME EXAMPLE
SQL REPLACE EXAMPLE
SQL DELETE EXAMPLE
SQL DROP EXAMPLE
SQL CREATE EXAMPLE
SQL EXPLAIN EXAMPLE
SQL SELECT EXAMPLE
SQL SHOW EXAMPLE
SQL USE EXAMPLE

SQL DESCRIBE EXAMPLE
SQL ANALYZE EXAMPLE
SQL HELP EXAMPLE
SQL INSERT EXAMPLE
SQL RESTORE EXAMPLE
SQL CECKSUM EXAMPLE
SQL CHECK EXAMPLE
SQL BACKUP EXAMPLE
SQL LOAD EXAMPLE
SQL UPDATE EXAMPLE
SQL DATA EXAMPLE
SQL CONSTRAINTS EXAMPLE
SQL KEY EXAMPLE
SQL FOREIGN_KEY EXAMPLE
SQL PRIMARY_KEY EXAMPLE
SQL TABLE EXAMPLE
SQL COLUMN EXAMPLE
SQL VIEW EXAMPLE
SQL STATEMENT EXAMPLE
SQL PERMISSION EXAMPLE
SQL INDEX EXAMPLE
SQL PASSWORD EXAMPLE
SQL FUNCTION EXAMPLE
SQL TRIGGER EXAMPLE
SQL USER EXAMPLE
SQL DATABASE EXAMPLE
SQL ROW EXAMPLE
SQL ALTER DESCRIPTION
SQL REVOKE DESCRIPTION
SQL GRANT DESCRIPTION
SQL TRUNCATE DESCRIPTION
SQL OPTIMIZE DESCRIPTION
SQL REPAIR DESCRIPTION
SQL SET DESCRIPTION
SQL MERGE DESCRIPTION
SQL RENAME DESCRIPTION
SQL REPLACE DESCRIPTION
SQL DELETE DESCRIPTION
SQL DROP DESCRIPTION
SQL CREATE DESCRIPTION
SQL EXPLAIN DESCRIPTION
SQL SELECT DESCRIPTION
SQL SHOW DESCRIPTION
SQL USE DESCRIPTION
SQL DESCRIBE DESCRIPTION
SQL ANALYZE DESCRIPTION
SQL HELP DESCRIPTION
SQL INSERT DESCRIPTION
SQL RESTORE DESCRIPTION
SQL CECKSUM DESCRIPTION
SQL CHECK DESCRIPTION
SQL BACKUP DESCRIPTION
SQL LOAD DESCRIPTION
SQL UPDATE DESCRIPTION
SQL DATA DESCRIPTION
SQL CONSTRAINTS DESCRIPTION
SQL KEY DESCRIPTION
SQL FOREIGN_KEY DESCRIPTION
SQL PRIMARY_KEY DESCRIPTION
SQL TABLE DESCRIPTION

SQL COLUMN DESCRIPTION
SQL VIEW DESCRIPTION
SQL STATEMENT DESCRIPTION
SQL PERMISSION DESCRIPTION
SQL INDEX DESCRIPTION
SQL PASSWORD DESCRIPTION
SQL FUNCTION DESCRIPTION
SQL TRIGGER DESCRIPTION
SQL USER DESCRIPTION
SQL DATABASE DESCRIPTION
SQL ROW DESCRIPTION
SQL ALTER MANUAL
SQL REVOKE MANUAL
SQL GRANT MANUAL
SQL TRUNCATE MANUAL
SQL OPTIMIZE MANUAL
SQL REPAIR MANUAL
SQL SET MANUAL
SQL MERGE MANUAL
SQL RENAME MANUAL
SQL REPLACE MANUAL
SQL DELETE MANUAL
SQL DROP MANUAL
SQL CREATE MANUAL
SQL EXPLAIN MANUAL
SQL SELECT MANUAL
SQL SHOW MANUAL
SQL USE MANUAL
SQL DESCRIBE MANUAL
SQL ANALYZE MANUAL
SQL HELP MANUAL
SQL INSERT MANUAL
SQL RESTORE MANUAL
SQL CECKSUM MANUAL
SQL CHECK MANUAL
SQL BACKUP MANUAL
SQL LOAD MANUAL
SQL UPDATE MANUAL
SQL DATA MANUAL
SQL CONSTRAINTS MANUAL
SQL KEY MANUAL
SQL FOREIGN_KEY MANUAL
SQL PRIMARY_KEY MANUAL
SQL TABLE MANUAL
SQL COLUMN MANUAL
SQL VIEW MANUAL
SQL STATEMENT MANUAL
SQL PERMISSION MANUAL
SQL INDEX MANUAL
SQL PASSWORD MANUAL
SQL FUNCTION MANUAL
SQL TRIGGER MANUAL
SQL USER MANUAL
SQL DATABASE MANUAL
SQL ROW MANUAL
SQL ALTER TUTORIAL
SQL REVOKE TUTORIAL
SQL GRANT TUTORIAL
SQL TRUNCATE TUTORIAL
SQL OPTIMIZE TUTORIAL

SQL REPAIR TUTORIAL
SQL SET TUTORIAL
SQL MERGE TUTORIAL
SQL RENAME TUTORIAL
SQL REPLACE TUTORIAL
SQL DELETE TUTORIAL
SQL DROP TUTORIAL
SQL CREATE TUTORIAL
SQL EXPLAIN TUTORIAL
SQL SELECT TUTORIAL
SQL SHOW TUTORIAL
SQL USE TUTORIAL
SQL DESCRIBE TUTORIAL
SQL ANALYZE TUTORIAL
SQL HELP TUTORIAL
SQL INSERT TUTORIAL
SQL RESTORE TUTORIAL
SQL CECKSUM TUTORIAL
SQL CHECK TUTORIAL
SQL BACKUP TUTORIAL
SQL LOAD TUTORIAL
SQL UPDATE TUTORIAL
SQL DATA TUTORIAL
SQL CONSTRAINTS TUTORIAL
SQL KEY TUTORIAL
SQL FOREIGN_KEY TUTORIAL
SQL PRIMARY_KEY TUTORIAL
SQL TABLE TUTORIAL
SQL COLUMN TUTORIAL
SQL VIEW TUTORIAL
SQL STATEMENT TUTORIAL
SQL PERMISSION TUTORIAL
SQL INDEX TUTORIAL
SQL PASSWORD TUTORIAL
SQL FUNCTION TUTORIAL
SQL TRIGGER TUTORIAL
SQL USER TUTORIAL
SQL DATABASE TUTORIAL
SQL ROW TUTORIAL

ODP Categories

Top/Computers/Programming/Languages/SQL

Configuration Info

Path = /home/slawless/occscrawl

BowPath = /usr/local/bin

HeritrixPath = /home/slawless/heritrix-1.4.0/bin

Topic = SQL

MaxURLCount = 0

Max-Search-Results = 1

topN = 200

Negative-Category-Count = 200

KEYWORD-FLAG = 1

ODP-FLAG = 1

CutOff = 0.9

Google-License-Key = ********************/********

Host = 134.226.35.161

Proxy = http://www-proxy.cs.tcd.ie:8080

Port = 5555

User = slawless

Pass = ********

Seeds

http://www.1keydata.com/sql/
http://beginner-sql-tutorial.com/
http://sqlzoo.net/
http://www.w3schools.com/sql
http://www.sql-tutorial.com/
http://docs.ingres.com/sqlref
http://www.sql.org/sql-database/sql-tutorial/
http://www.sql.org/sql-database/
http://sqlcourse.com/
http://www.sqlsnippets.com/en/home.html
http://www.firstsql.com/tutor.htm
http://www.firstsql.com/tutor3.htm
http://www.firstsql.com/tutor6.htm
http://dev.mysql.com/doc/refman/5.0/en/sql-syntax.html
http://dev.mysql.com/doc/refman/6.0/en/sql-syntax.html
http://databases.about.com/od/sql/u/learning_sql.htm
http://www.rawos.com/odbc/sqlguide/
http://www.1keydata.com/sql/sql-alter-table.html
http://www.rawos.com/odbc/sqlguide/chapters/rights.htm
http://beginner-sql-tutorial.com/sql-grant-revoke-privileges-roles.htm
http://www.1keydata.com/sql/sqltruncate.html
http://www.basis.com/support/tips/sqloptimization.html
http://en.wikipedia.org/wiki/MERGE
http://www.oracle-base.com/articles/9i/RenamingColumnsAndConstraints9i.php
http://searchoracle.techtarget.com/expert/KnowledgebaseAnswer/0,289625,sid41_gci1321496,00.html
http://www.1keydata.com/sql/sqldelete.html
http://www.1keydata.com/sql/sqldrop.html
http://www.adp-gmbh.ch/ora/sql/create_table.html
http://php.about.com/od/mysqlcommands/MySQL_Commands.htm
http://www.1keydata.com/sql/sql.html
http://beginner-sql-tutorial.com/sql-insert-statement.htm
http://www.orafaq.com/wiki/SQL*Loader_FAQ
http://www.sql-tutorial.com/sql-update-sql-tutorial/
http://www.1keydata.com/sql/sql.html
http://www.1keydata.com/sql/sql-constraint.html
http://en.wikipedia.org/wiki/Foreign_key
http://en.wikipedia.org/wiki/Primary_key
http://www.1keydata.com/sql/sqltruncate.html
http://www.adp-gmbh.ch/ora/sqlplus/column.html
http://www.sql-tutorial.com/sql-views-sql-tutorial/

http://www.firstsql.com/tutor2.htm
http://www.sql-tutorial.com/sql-indexes-sql-tutorial/
http://msdn.microsoft.com/en-us/library/44xx6c68(VS.80).aspx
http://www.postgresql.org/docs/8.1/interactive/sql-revoke.html
http://www.faqs.org/docs/ppbook/x19270.htm
http://www.1keydata.com/sql/sqltruncate.html
http://msdn.microsoft.com/en-us/library/yey80zw6(VS.80).aspx
http://beginner-sql-tutorial.com/sql-alter-table.htm
http://www.dbmaker.com.tw/reference/manuals/sql/contents.html
http://www.dbmaker.com.tw/reference/manuals/sql/functions/replace.html
http://www.comptechdoc.org/independent/database/begin/sqldelete.html
http://www.1keydata.com/sql/sqldrop.html
http://msdn.microsoft.com/en-us/library/h09t6a82(VS.80).aspx
http://www.comptechdoc.org/independent/database/begin/sqlselect.html
http://php.about.com/od/mysqlcommands/g/show_tables.htm
http://manuals.sybase.com/onlinebooks/group-as/asg1250e/sqlug/@Generic__BookTextView/18240;pt=17998
http://php.about.com/od/mysqlcommands/g/describe_table.htm
http://www.1keydata.com/sql/sql.html
http://www.comptechdoc.org/independent/database/begin/sqlinsert.html
http://www.1keydata.com/sql/sqlupdate.html
http://infolab.stanford.edu/~ullman/fcdb/oracle/or-nonstandard.html
http://vista.intersystems.com/csp/docbook/DocBook.UI.Page.cls?KEY=GSQL
http://vista.intersystems.com/csp/docbook/DocBook.UI.Page.cls?KEY=RSQL
http://dbaforums.org/oracle/index.php?showtopic=16610
http://www.comp.nus.edu.sg/~ooibc/courses/sql/
http://www.comp.nus.edu.sg/~ooibc/courses/sql/ddl_table.htm
http://ocw.mit.edu/NR/rdonlyres/Urban-Studies-and-Planning/11-521Spatial-Database-Management-and-Advanced-Geographic-Information-SystemsSpring2003/4EABB334-E7C0-4A12-9AE7-96A7F696BF9F/0/sqlnotes.pdf
http://msdn.microsoft.com/en-us/library/h7y2325d(VS.80).aspx
http://www.baycongroup.com/sql_command_reference.htm
http://msdn.microsoft.com/en-us/library/aa977477(VS.71).aspx
http://www.dbmaker.com.tw/reference/manuals/sql/contents.html
http://msdn.microsoft.com/en-us/library/xtydh4db(VS.80).aspx
http://www.comptechdoc.org/independent/database/mysql/sqluser.html
http://msdn.microsoft.com/en-us/library/h09t6a82(VS.80).aspx
http://www.exforsys.com/tutorials/programming-concepts/sql-basic-concepts.html
http://www.cs.bc.edu/~sciore/papers/IFIP00.pdf
http://www.geekinterview.com/question_details/35094
http://www.patentstorm.us/patents/6996557-description.html
http://conceptfortheday.blogspot.com/2007/11/sql-set-option-statement.html
http://www.geekinterview.com/question_details/2488
http://conceptfortheday.blogspot.com/2008/09/use-sql-to-remove-extra-spaces.html
http://www.exforsys.com/tutorials/programming-concepts/sql-basic-concepts.html
http://conceptfortheday.blogspot.com/2008/08/v5r3-enhancement-in-sql.html
http://www.training-classes.com/programs/00/35/3574_oracle_sql_basic_select_statements.php
http://www.cs.toronto.edu/~libkin/csc343/f04/set2.2up.pdf
http://www12.georgetown.edu/scs/ccpe/courses/introduction_to_oracle_sql_10g.cfm
http://conceptfortheday.blogspot.com/2008/10/conditional-insert-in-sql.html
http://www.informatik.uni-bonn.de/~behrend/btw_paper.pdf
http://books.google.com/books?id=Q5cc9DCiov4C&pg=PA35&lpg=PA35&dq=%2B%2BSQL+FOREIGN_KEY+CONCEPT+--SQLSERVER+--%22SQL+SERVER%22+--asp+--java&source=bl&ots=jEDI_CRfcp&sig=ccfxO8vTZZcc_www5Aug7hIdKm4&hl=en
http://www.expertwebinstalls.com/cgi_tutorial/basic_relational_database_concepts.html
http://www.exforsys.com/tutorials/programming-concepts/sql-basic-concepts.html
http://search400.techtarget.com/search400/downloads/SQL_concepts.pdf
http://www.expertwebinstalls.com/cgi_tutorial/basic_relational_database_concepts.html
http://conceptfortheday.blogspot.com/2008/10/conditional-insert-in-sql.html
http://www.devhood.com/Tutorials/tutorial_details.aspx?tutorial_id=156
http://mo.co.za/open/sqlrevoke.pdf

http://archives.postgresql.org/pgsql-php/2002-01/msg00061.php
http://dmiessler.com/blog/how-does-one-explain-sql-injection-to-a-non-techie
http://www.nabble.com/Help-with-constructing-a-SQL-query-td14129309.html
http://mail.python.org/pipermail/python-list/2005-December/355417.html
http://www.sir.com.au/help/sql_menu.htm#select
http://infolab.stanford.edu/~ullman/fcdb/oracle/or-triggers.html
http://www.sqlalchemy.org/docs/03/sqlconstruction.html
http://publib.boulder.ibm.com/infocenter/db2luw/v9/topic/com.ibm.datatools.sqlwizard.doc/topics/ruisqlwiznotebook.html
http://msdn.microsoft.com/en-us/library/bb738573.aspx
http://beginner-sql-tutorial.com/sql-alter-table.htm
http://www.postgresql.org/docs/8.3/interactive/sql-commands.html
http://www.adp-gmbh.ch/ora/sqlplus/set_define.html
http://publib.boulder.ibm.com/infocenter/db2luw/v9/topic/com.ibm.help.etl.doc/designing/data_flow/toptgtmerge.html
http://searchoracle.techtarget.com/expert/KnowledgebaseAnswer/0,289625,sid41_gci1321496,00.html
http://en.wikipedia.org/wiki/Data_Definition_Language
http://www.1keydata.com/sql/sqldrop.html
http://php.about.com/od/mysqlcommands/g/describe_table.htm
http://beginner-sql-tutorial.com/sql-insert-statement.htm
http://www.network-theory.co.uk/docs/postgresql/vol1/pg_restore.html
http://www.sql-tutorial.com/sql-update-sql-tutorial/
http://en.wikipedia.org/wiki/Data_Definition_Language
http://vpf-web.harvard.edu/applications/ad_hoc/key_functions_in_oracle_sql.pdf
http://en.wikipedia.org/wiki/Primary_key
http://www.sql.org/sql-database/postgresql/manual/ddl.html
http://support.alphasoftware.com/alphafivehelp/Shared_Pages/Define_SQL_Statement_Dialog.htm
http://www.csis.ul.ie/Modules/cs4513/chapter15.pdf
http://docs.hp.com/cgi-bin/doc3k/B3621690086.12555/18
http://www.1keydata.com/sql/sql-alter-table.html
http://www.cs.umbc.edu/help/oracle8/server.815/a67779/ch4k.htm
http://developer.postgresql.org/docs/postgres/sql-grant.html
http://www.1keydata.com/sql/sqltruncate.html
http://www.basis.com/support/tips/sqloptimization.html
http://www.itl.nist.gov/div897/ctg/dm/sql_examples.htm
http://www.cs.utexas.edu/users/cannata/dbms/SQL%20Merge.html
http://www.mydigitallife.info/2007/04/23/how-to-find-and-replace-text-in-mysql-database-using-sql/
http://en.wikipedia.org/wiki/DELETE
http://www.1keydata.com/sql/sqldrop.html
http://sql-info.de/mysql/examples/CREATE-TABLE-examples.html
http://industrex.org/dynamic/reference/sql/sql-help/
http://msdn.microsoft.com/en-us/library/h54fa37c(VS.80).aspx
http://www.itl.nist.gov/div897/ctg/dm/sql_examples.htm
http://www.1keydata.com/sql/sqlupdate.html
http://www.1keydata.com/sql/sql-constraint.html
http://vpf-web.harvard.edu/applications/ad_hoc/key_functions_in_oracle_sql.pdf
http://en.wikipedia.org/wiki/Foreign_key
http://en.wikipedia.org/wiki/Primary_key
http://sql-info.de/mysql/examples/CREATE-TABLE-examples.html
http://www.geocities.com/SiliconValley/Vista/2207/sql5.html
http://philip.greenspun.com/sql
http://www.baycongroup.com/sql_database_tutorial.htm
http://www.peachpit.com/articles/article.aspx?p=30681&seqNum=6
http://www.1keydata.com/sql/sql-alter-table.html
http://beginner-sql-tutorial.com/sql-grant-revoke-privileges-roles.htm
http://www.1keydata.com/sql/sqltruncate.html
http://www.basis.com/support/tips/sqloptimization.html
http://en.wikipedia.org/wiki/MERGE
http://www.1keydata.com/sql/sqldelete.html
http://www.1keydata.com/sql/sqldrop.html

http://www.industrex.com/dynamic/reference/sql/sql-help/sqleoper.htm
http://seagullproject.org/forum/index.php?goto=1392&t=msg
http://www.1keydata.com/sql/sql-constraint.html
http://en.wikipedia.org/wiki/Primary_key
http://php.about.com/od/mysqlcommands/g/describe_table.htm
http://beginner-sql-tutorial.com/sql-insert-statement.htm
http://beginner-sql-tutorial.com/sql-alter-table.htm
http://beginner-sql-tutorial.com/sql-grant-revoke-privileges-roles.htm
http://beginner-sql-tutorial.com/sql-grant-revoke-privileges-roles.htm
http://www.1keydata.com/sql/sqltruncate.html
http://dbis.ucdavis.edu/courses/sqltutorial
http://tech.inhelsinki.nl/2007-01-27/
http://www.mydigitallife.info/2007/04/23/how-to-find-and-replace-text-in-mysql-database-using-sql/
http://php.about.com/od/learnmysql/ss/create_tables.htm
http://dbis.ucdavis.edu/courses/sqltutorial/tutorial.pdf
http://www.sql-tutorial.com/sql-order-by-sql-tutorial/
http://www.analysisandsolutions.com/code/mysql-tutorial.htm
http://industrex.org/dynamic/reference/sql/sql-help/
http://www.sql-tutorial.com/sql-insert-sql-tutorial/
http://www.1keydata.com/sql/sqlupdate.html
http://voboghurey.blogspot.com/2008/07/foreign-key-my-sql-example.html
http://beginner-sql-tutorial.com/sql-integrity-constraints.htm
http://www.baycongroup.com/sql_database_tutorial.htm
http://www.sqlsnippets.com/en/topic-12180.html
http://www.sql-tutorial.com/sql-views-sql-tutorial/
http://www.1keydata.com/sql/sql.html
http://www.sql-tutorial.com/sql-indexes-sql-tutorial/
http://www.milw0rm.com/papers/202
http://www.exforsys.com/tutorials/oracle-9i/sql-functions.html
http://ocw.mit.edu/NR/rdonlyres/Urban-Studies-and-Planning/11-521Spatial-Database-Management-and-Advanced-Geographic-Information-SystemsSpring2003/331EBBDF-AFB9-47F1-87B7-565B09A4C500/0/lab1.pdf
http://ocw.mit.edu/NR/rdonlyres/Electrical-Engineering-and-Computer-Science/6-830Fall-2005/47F39F93-119E-42C0-8013-FF1336F8FAA5/0/ps1.pdf
http://ocw.mit.edu/NR/rdonlyres/Sloan-School-of-Management/15-564Spring2003/967BFD33-30C1-4A1F-8ED2-98C16A244FA4/0/sql_query.pdf

Appendix F – U-CREATe Trial Documents

Pre-Trial Test

U-CREATe

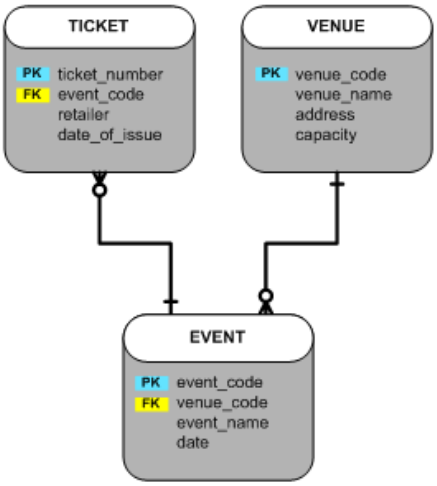
User-driven Content Retrieval, Exploration and Assembly Toolkit for eLearning

Pre-Trial Questionnaire

Name:

Please Note: This questionnaire is to be carried out in isolation. Please turn off your monitor and do not confer. This is a learning exercise; the following questions are to examine the performance of the U-CREATe tool and are not for assessment purposes. Please attempt all questions and answer as honestly as possible, thanks ☺

The following is a basic database diagram for a ticketing system:



Question 1:

(i) Write the SQL statement required to create a database table called "EVENT". The table should contain the following fields

event_code – A number between 1 and 9999
venue_code – A number between 1 and 9999
event_name – Text detailing the name of each event
date – The date the event takes place

(ii) What would need to be added to the statement you created in part A to include the event_code field as the Primary Key of the table.

(iii) What would need to be added to the statement you created in part A to include the venue_code field as a Foreign Key of the table. The venue_code Foreign Key can be found in the table called VENUE.

Question 2:

(i) Write the SQL statement required to insert a record into the database table "EVENT". The values to be inserted are as follows

event_code – 1
venue_code – 2
event_name – Snow Patrol
date – 28/07/2008

(ii) Show how the statement you created in part A would need to be altered if the value to be inserted into venue_code is provided by a record in the table "VENUE" where the field venue_name is equal to "Point Depot".

(iii) Show how the statement you created in part A would need to be altered if the value to be inserted into venue_code is provided by the record in the table "VENUE" with the greatest venue_code.

Question 3:

(i) Write the SQL statement required to return all records from the database table "EVENT" where the event_code is less than 10 and the event_name is "Snow Patrol".

(ii) Write the SQL statement required to return all the event_codes from the database table "EVENT" where the venue is "Point Depot".

(iii) Write the SQL statement required to return all the event_codes and event_names from the database table "EVENT" where the venue capacity is 6000 and the event occurs in 2008.

Post-Trial Test

U-CREATe

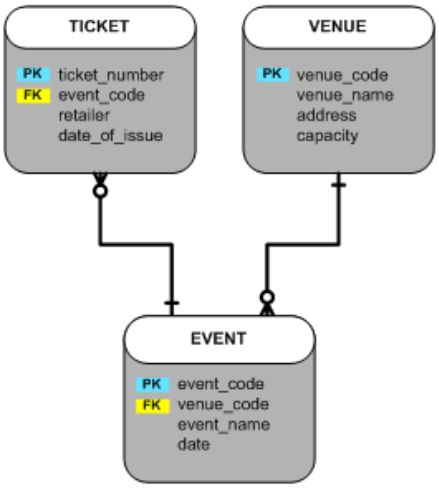
User-driven Content Retrieval, Exploration and Assembly Toolkit for eLearning

Post-Trial Questionnaire

Name:

Please Note: This questionnaire is to be carried out in isolation. Please turn off your monitor and do not confer. This is a learning exercise; the following questions are to examine the performance of the U-CREATe tool and are not for assessment purposes. Please attempt all questions and answer as honestly as possible, thanks ☺

The following is a basic database diagram for a ticketing system:



Question 1:

(i) Write the SQL statement required to create a database table called "TICKET". The table should contain the following fields

ticket_code – A number between 1 and 9999
event_code – A number between 1 and 9999
retailer – Text detailing the name of the ticket seller
date_of_issue – The date the ticket was issued

(ii) What would need to be added to the statement you created in part A to include the ticket_code field as the Primary Key of the table.

(iii) What would need to be added to the statement you created in part A to include the event_code field as a Foreign Key of the table. The event_code Foreign Key can be found in a table called EVENT.

Question 2:

(i) Write the SQL statement required to insert a record into the database table "TICKET". The values to be inserted are as follows

ticket_code – 100
event_code – 1
retailer – Ticketmaster
date_of_issue – 20/01/2008

(ii) Show how the statement you created in part A would need to be altered if the value to be inserted into event_code is provided by a record in the table "EVENT" where the field event_name is equal to "Snow Patrol".

(iii) Show how the statement you created in part A would be altered if the value to be inserted into event_code is provided by the record in the table "EVENT" with the greatest event_code.

Question 3:

(i) Write the SQL statement required to return all records from the database table "TICKET" where the ticket_code is less than 100 and the retailer is "Ticketmaster".

(ii) Write the SQL statement required to return all the ticket_codes from the database table "TICKET" where the event is "Snow Patrol".

(iii) Write the SQL statement required to return all the ticket_codes and retailers from the database table "TICKET" where the venue code is 100 and the date of issue of the ticket was in 2008.

U-CREATe exercise tasks

SQL Exercise - ST3001

Using U-CREATe, you are required to create a mind map that shows how SQL is used to create, populate and query a database. A diagram of the database you will be asked to create is available below. You can browse the OCCS at any stage during the to find information to help you along the way with creating the required SQL statements.

In the U-CREATe Mindmap that you have been provided, you will see two nodes on the left and three on the right. The two nodes on the left link to this website and to the database diagram. The three nodes on the right are where you should begin adding your own nodes to the map. To add a new node to the Mindmap, right click on an existing node and click "Add Child Node" or click on a node and press insert on your keyboard. **Please enter an entire SQL statement into each node you create.** So for Question 1, you should end up with three of your own nodes under "Database Creation Statements".

(Section 1) Under the node "Database Creation Statements" please enter a new child node for each of the SQL statements required to create the database described in the diagram provided. To help you in the construction of these statements you can browse the OCCS for content relating to database creation and SQL statements. The OCCS can also provide information on neccessary contraints and clauses. On each node that you create in the U-CREATe Mindmap, please add a hyperlink to each page that you found helpful during the creation of that node. To do this, right click on the node that you created and click "Add a Link to OCCS Content" then cut and paste the hyperlink from the OCCS content.

(Section 2) Under the node "Database Population Statements" please enter a new child node for each of the statements required to populate the database described in the schema diagram with the following content.

Staff Members
1, John O'Toole, DOB: 28/07/1970, Phone: 01-2864328.
2, Eleanor Rigby, DOB: 05/08/1966, Phone: 01-8961222.

Customers
1, John Ewing, DOB: 11/01/1975, Phone: 01-2862222.
2, Julius Hibbert, DOB: 19/04/1987, Phone: 01-8961111.

Lessons
1, John Ewing, Eleanor Rigby, 28/10/2007, EUR40.
2, Julius Hibbert, John O'Toole, 01/11/2007, EUR30.

To help you in the construction of these statements you can browse the OCCS for content relating to database population. On each node in the U-CREATe map, please add a hyperlink to each relevant piece of learning content that you found helpful during the creation of that node. To do this, right click on the node that you created and click "Add a Link to OCCS Content" then cut and paste the hyperlink from the OCCS content.

(Section 3) Under the node "Database Query Statements" please enter a new child node for each of the statements required to extract the following information from the database.

Query A
On what dates has Eleanor Rigby given Lessons?

Query B
What is the total cost of the Lessons taken by John Ewing?

Query C
What was the maximum cost of a lesson in this Driving School?

To help you in the construction of these statements you can browse the OCCS for content relating to database querying. The OCCS can also provide information on any SQL functions that are required. On each node in the U-CREATe map, please add a hyperlink to each relevant piece of learning content that you found helpful during the creation of that node. To do this, right click on the node that you created and click "Add a Link to OCCS Content" then cut and paste the hyperlink from the OCCS content.

When you have completed the trial please save your Mindmap as username.mm and drop it into the PUT folder or mail it to seamus.lawless@cs.tcd.ie Then inform me and I can give you the Post-Trial and Usability Questionnaire to finish up, Thanks!

SUS Questionnaire

U-CREATe

User-driven Content Retrieval, Exploration and Assembly Toolkit for eLearning

SUS Questionnaire

Name:

Please Note: This questionnaire is to examine the performance and usability of both the U-CREATe interface and the OCCS. Please answer as honestly as possible, thanks ☺

	Strongly Disagree				Strongly Agree
1. I think that I would like to use this system frequently.					
2. I found the system unnecessarily complex.					
3. I thought the system was easy to use.					
4. I think that I would need the support of a technical person to be able to use this system					
5. I found the various functions in this system were well integrated					
6. I thought there was too much inconsistency in this system					
7. I would imagine that most people would learn to use this system very quickly.					
8. I found the system very cumbersome to use.					
9. I felt very confident using the system.					
10. I needed to learn a lot of things before I could get going with this system.					

Context-specific usability questionnaire

U-CREATe

User-driven Content Retrieval, Exploration and Assembly Toolkit for eLearning

Usability Questionnaire

Name:

Please Note: This questionnaire is to examine the performance and usability of both the U-CREATe interface and the OCCS. Please answer as honestly as possible, thanks ☺

How useful did you find this as an educational experience?

Not at all Useful									Very Useful
1	2	3	4	5	6	7	8	9	10

Do you believe this process would be beneficial in supporting the learning of a subject area?

Not Beneficial									Very Beneficial
1	2	3	4	5	6	7	8	9	10

How easy did you find the U-CREATe interface to use?

Very Difficult									Extremely Easy
1	2	3	4	5	6	7	8	9	10

Are there any improvements that you would suggest making to the U-CREATe interface?

Was it difficult to find relevant content in the OCCS?

Very Difficult									Extremely Easy
1	2	3	4	5	6	7	8	9	10

Was the OCCS interface intuitive to use?

Very Difficult									Extremely Easy
1	2	3	4	5	6	7	8	9	10

Are there any improvements that you would suggest making to the OCCS interface?

Any other comments?

Appendix G – Author Publications

Levacher, K., Hynes, E., Lawless, S., O'Connor, A. & Wade, V. "Towards a Framework for Open-Corpus Localisation Supporting Adaptive Hypermedia". In the Proceedings of the International Workshop on Dynamic and Adaptive Hypertext, DAH'09, at Hypertext 2009, Torino, Italy. June 29th, 2009.

Steichen, B., Lawless, S. & Wade, V. "*Dynamic Hypertext Generation for Reusing Open Corpus Content"*, In the Proceedings of the 20th ACM Conference on Hypertext and Hypermedia, Hypertext 2009, Torino, Italy. 29th June – 1st July, 2009.

Lawless, S., Hederman, L., Wade, V. "*Enhancing Access to Open Corpus Educational Content: Learning in the Wild*", In the Proceedings of the 21st ACM Conference on Hypertext and Hypermedia, Hypertext 2008, Pittsburgh, PA, U.S.A. June 19th-21st, 2008.

Lawless, S., Hederman, L., Wade, V. "*OCCS: Enabling the Dynamic Discovery, Harvesting and Delivery of Educational Content from Open Corpus Sources*", In the Proceedings of the 8th IEEE International Conference on Advanced Learning Technologies, I-CALT 2008, Santander, Cantabria, Spain. July 1st-5th, 2008.

Dagger, D., O'Connor, A., Lawless, S., Walsh, E., Wade, V. "*Service Oriented TEL Platforms: From Monolithic Systems to Flexible Services*", In IEEE Internet Computing, Special Issue on Distance Learning, V. Wade & H. Ashman (eds.), vol. 11(3), pp. 28-35. May-June, 2007.

Lawless, S. "Open Corpus Learning Content: Harvesting Knowledge to provide Equitable Access to Education for All". In the 2nd International Education Without Borders Conference, EWB2007, Abu Dhabi, United Arab Emirates. February 25th-27th, 2007.

Lawless, S. & Wade, V. "*Dynamic Content Discovery, Harvesting and Delivery, from Open Corpus Sources, for Adaptive Systems*", In the Proceedings of the Fourth International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems, AH 2006, V. Wade, H. Ashman, B. Smyth (eds.), Dublin, Ireland, LNCS 4018, Springer-Verlag, pp. 445–451. June 20th-23rd, 2006.

Lawless, S., Dagger, D., Wade, V. "*Towards Requirements for the Dynamic Sourcing of Open Corpus Learning Content*", In the Proceedings of the World Conference on E-Learning in Corporate, Government, Healthcare, and Higher Education, E-Learn 2006, Honolulu, Hawaii, USA, T.C. Reeves & S.F. Yamashita (eds.). October 13th-17th, 2006.

Lawless, S., Wade, V., Conlan, O. "*Dynamic Contextual TEL - Dynamic Content Discovery, Capture and Learning Object Generation from Open Corpus Sources*", In the Proceedings of the World Conference on E-Learning in Corporate, Government, Healthcare and Higher Education, E-Learn 2005, Vancouver, B.C., G. Richards (ed.), AACE, pp. 2158 – 2165. October 24th-28th, 2005.