Comparing SpamAssassin with CBDF email filtering

Cormac O'Brien[†] **& Carl Vogel**[†‡]
Computational Linguistics Group[†] & Centre for Computing and Language Studies[‡]
Trinity College, University of Dublin
{obrience,vogel}@tcd.ie

Abstract

In this paper, we compare the email filtering software SpamAssassin with a statistical email filter, known as *chi by degrees of freedom*. We examine SpamAssassin's filtering techniques and ascertain their effectiveness. By comparing the two filtering methods, we will show that a rule based filter such as SpamAssassin is in fact the wrong approach for something as inconstant as unsolicited bulk email.

1 Introduction

SpamAssassin[1] (SA, henceforth) is a now popular mail filter, which uses text analysis to identify spam. It is very popular in unix circles, and is even used by the author's home institution, Trinity College, Dublin. The filter uses a giant rule base to examine an incoming mail. Failure to comply with a certain rule will result in the mail having points assigned to it. If the mail amasses more than 5.0 points, then it is classified as spam. SA analyses both the header of the mail and its body. Also the filter takes advantage of blacklists[2] such as *mail-abuse.org*. As well as this, SA utilizes Vipul's Razor, which is a collaborative spam-tracking database. SA is freely available, easy to configure, and very flexible. One advantage of SA is that it clearly indicates to the user why the mail has been marked as spam by explaining exactly which rules have been violated. This makes it easier to spot why false positives have been so marked. However, as we will discuss later, a large number of false positives are an unpleasant feature of SA. False positives (legitimate emails which are marked as spam) are a huge concern in spam filtering. Mehran Sahami and Horvitz (1998) concluded that a false positive is 999 times worse than a false negative (spam mail marked as legitimate). In this paper we will look at the reasons for this. The *chi by degrees of freedom* test (CBDF, Henceforth) is used in the field of authorship identification. Chaski (1998) reviews many of the current techniques for language-based author identification. She identifies character-based techniques as the most effective. It was first proposed by Kilgarriff and Salkie (1996). While Van Gijsel (2002) used it to automatically classify political manifestos, it was first proposed as a possible method for filtering email by O'Brien and Vogel (2003b). One argument for using authorship identification methods to filter spam emails is provided by Ludlow (2002). He stated that the vast majority of spam emails are the work of just 150 spammers worldwide. In section 2 we examine SA more closely. Section 3 presents the CBDF method of email filtering. In section 4 we examine the filtering results of SA and CBDF. We will examine the reason behind some of the false positives in section 5. Finally conclusions will be drawn in section 6.

2 SpamAssassin

SpamAssassin uses a wide range of heuristic tests on emails to identify spam. These tests will assign a score to the email. Each test can assign anything from 0.001 points to 4.5 points. SA requires 5.0 points for a mail to be considered spam. Some of SA's tests include '*HTML mail with non-white background*' (0.3 points), '*HTML comments which obfuscate text*' (2.1 points) and '*contains the line: As seen on national TV!*' (1.4 points). It is possible for the user to change the number of points that SA assigns for each test. Also, the user can disable

[1] www.spamassassin.org, Last verified 29/09/2003

[2] A blacklist is a list of know email spammers. If the incoming mail is from someone on a blacklist, it is assumed to be spam

a test to combat false positives.

Some of these tests are used to analyse the header of the mail. Spammers will often try to hide their identity by changing information on the header. Forging a header is effective for tricking the untrained email reader. However, it is relatively easy for a filter to spot these forgeries. '*To: is empty*' (1.994 points) is an example of a test that examines the header. If the 'To:' line in the header is empty, then the mail is assigned 1.994 points.

Most of SA's tests examine the body of the email. An example would be '*Contains word 'guarantee' in all-caps*' (1.123 points). The problem with this type of key-word filtering is that it becomes obsolete very quickly. Spammers tend to own spam filtering software themselves. They will send a spam mail to themselves over and over again, tweaking it each time until it is finally accepted by the filter as a legitimate email. Then the spammer knows that his spam mail will beat the filter, and he sends it en masse to his list of harvested email addresses. A spammer will soon ascertain that his emails containing the word 'GUARANTEE' are not getting through. In fact, SpamAssassin have been kind enough to place all their tests on the Internet[3]. This reads like a step-by-step guide to beating spam filters. It is probable that many spammers examine this web-page before composing their spam mails.

SA also uses blacklists to boost it's performance. Blacklists are a very effective way of filtering spam without looking at the content of the email in question. The filter simply checks the sender's address. If that address is on the blacklist, then it is considered spam. However, this method is not without its problems. Spammers often forge email address to make it look like the email has come from an innocuous sender. This may lead to that sender having his address placed on a blacklist unbeknownst to him.

Below we can see an example of a mail that has been categorized as spam by SpamAssassin. It is a common Nigerian 419 scam[4]. The content of the email has been left out by the author for brevity. As can be seen from line 4, SA attaches "*****SPAM*****" to the beginning of a subject line if the mail has been marked as spam. This makes it easier for the user to filter marked spam into a separate folder. Attached to the beginning of the email text is line notifying the reader that the mail is spam (lines 6-8). This is followed by a content analysis of the mail (lines 10-20). The content analysis is a step by step guide for the reader explaining exactly why the mail has been marked as spam. This makes it easier for the reader to discover why false positives have been incorrectly categorized. In this example, we can see that the mail has received 17.60 points. SA only requires a mail to have 5.0 points to be considered spam. This high number is unsurprising for a 419 scam. Most of the points come from the sender forging the email to look like it came from MS Outlook Express (6.4 points in total). Other indicators include the sender's address ending in numbers (0.7 points) and the subject having a lot of exclamation marks (1.7 points)

```
1   Date: Wed, 1 Oct 2003 01:51:49 -0700
2   From: Tijani Ahmed <ahmedtijani87@post.cz>
3   To: cormac.obrien@cs.tcd.ie
4   Subject: *****SPAM***** ASSISTANCE & OFFER !!!
5
6   This mail is probably spam.  The original message has been attached along
7   with this report, so you can recognize or block similar unwanted mail
8   in future.
9
10   Content analysis details:   (17.60 points, 5 required)
11  FROM_ENDS_IN_NUMS  (0.7 points)  From: ends in numbers
12  RATWARE_OE_MALFORMED (2.9 points)  X-Mailer contains malformed Outlook Express
13  URGENT_BIZ         (0.2 points)  BODY: Contains urgent matter
14  DEAR_SOMETHING     (2.6 points)  BODY: Contains 'Dear (something)'
15  RISK_FREE          (0.9 points)  BODY: Risk free.  Suuurreeee....
16  SUBJ_ALL_CAPS      (1.1 points)  Subject is all capitals
17  DATE_IN_FUTURE_06_12 (1.3 points)  Date: is 6 to 12 hours after Received: date
18  FORGED_MUA_OUTLOOK (3.5 points)  Forged mail pretending to be from MS Outlook
19  PLING_PLING        (1.7 points)  Subject has lots of exclamation marks
20  NIGERIAN_BODY      (2.7 points)  Message body has multiple indications of Nigerian
spam
```

3 Authorship Identification in email filtering

According to Ludlow (2002), just 150 spammers may be responsible for the vast majority of spam that we receive. Baayen et al. (2002) produced research that indicated that "authors may have textual fingerprints, at least for texts produced by writers who are not consciously changing their style of writing across texts". If this is the case, we could use authorship identification (AID) methods to identify the textual fingerprints of spammers. This information could eliminate much of the spam that we receive today.

It would be possible for these spammers to consciously change their writing style when

[3] http://www.spamassassin.org/tests.html

[4] see http://easyweb.easynet.co.uk/ gcaselton/spam/nigerian.html, last verified 01/10/2003

sending spam emails, but we propose that the AID filter would learn as it filtered. Both spam and legitimate emails would be added to the filters training corpus. One of the advantages of this approach is that it becomes very difficult for spammers to knowingly write e-mail that can beat the filter. As the spammer will have no idea what is in each individuals training corpus, they will not be able to 'tweak' their mails until they beat the filter.

The AID method that we present is the *chi by degrees of freedom* approach. This method was used to by O'Brien and Vogel (2003b) to successfully filter email. To carry out the CBDF experiments we availed of software written by McCombe (2002). Van Gijsel (2002) has shown these programs to be effective for analyzing European right-wing party manifestos and O'Brien and Vogel (2003a) used them to explore the authorship of a poem attributed to Shakespeare. The programs are trained on files containing known spam and legitimate emails. The training files are concordanced: each file is indexed by its n-grams, with frequency counts. The n-grams can be of characters or words depending on what the user specifies. The user also specifies the type of n-gram to use (unigrams, bigrams, etc.). Then the program calculates the similarity value between the newly created files in terms of n-gram frequencies. To do this it carries out the CBDF test. This test was proposed by Kilgarriff and Salkie (1996). It is calculated by dividing the chi-square test value by the number of its degrees of freedom (number of n-grams minus one). The χ^2 test[5] tests the validity of the null hypothesis. The null hypothesis states that the difference between two sets of data is merely due to chance. Significance is a measure of the chance that rejecting the null hypothesis is wrong.

Unfortunately, with the χ^2 as-is the larger the sample size, the more likely the null hypothesis will be rejected. McCombe (2002, pg. 31) notes, "This was independently rediscovered when working on bigrams and trigrams and finding that very large numbers of them exhibited significant differences even in text samples written by the same author". Therefore, the CBDF test is used. The programs return a table of the similarity scores of each of the corpus pairs, whence it may be deduced whether the email in question is to be classified as 'spam' or 'legitimate'.

[5]For an in-depth explanation of the χ^2 test and an example of its use in corpus linguistics, see Oakes (1998)

4 Filtering Results

SpamAssassin's filtering results can be seen in table 1. Of the 3834 emails in the author's inbox, 188 were marked as spam, while 3646 were marked as legitimate. Of the 188 emails marked by SA as spam, 50 were false positives. This is 26% of the marked mails. For the user, it is far worse to receive a false positive, than a false negative. Having a spam email marked as legitimate can be an inconvenience, but having a legitimate email marked as spam can have serious consequences. Mehran Sahami and Horvitz (1998) say that their classifier has to be 99.9% certain that a mail is spam before it is marked as such. This is akin to saying that allowing one legitimate mail to be marked as spam is as bad as allowing 999 spam mails to be marked as legitimate. SA has marked 1.3% of all the authors legitimate emails as spam. This is an unacceptably high figure.

Looking at table 1, the statistics seem to be quite impressive. However, on closer inspection we see that of the total of 205 spam emails in the authors inbox, SA correctly identified only 138 or 67% of them. This is a very low figure for a spam filter. For example, Graham (2002) claims that his filter will correctly identify 99.5% of all spam, and will produce no false positives.

The CBDF results were much more promising. We trained the filter using a corpus of 1000 legitimate emails and 200 spam emails. The ratio between legitimate and spam emails does not have a noticeable effect on the filter (A ratio of 1:1 would provide no better accuracy than the ratio of 5:1 which we use. The larger the training corpus is, the more accurate the filter will be). The filter was then tested on 171 emails. 57 of these mails were spam and 114 mails were legitimate. The filter provided excellent results, with all 57 spam mails correctly identified. Even more promising was that the filter produced no false positives. All 114 of the legitimate emails were correctly marked.

Total Emails	Marked Legit	Marked Spam	False Pos.	False Neg.
3834	3646	188	50	67
100%	95%	5%	1.3%	1.75%

Table 1: Results of SpamAssassin filter

Total Emails	Marked Legit	Marked Spam	False Pos.	False Neg.
171	114	57	0	0
100%	100%	100%	0%	0%

Table 2: Results of 'Chi by Degrees of Freedom' filter

5 False Positives

One of the key issues in spam filtering is to avoid false positives. A false positive is a legitimate email that has been marked as spam. One can imagine that loosing a legitimate email is far more frustrating than receiving a spam mail in your inbox. Mehran Sahami and Horvitz (1998) concluded that it would be 999 times worse to loose a legitimate email than to receive a spam email. In this section we will examine some of the reasons that legitimate emails were incorrectly categorized. One of the false positives can be seen below.

```
1   Date: Tue, 29 Jul 2003 22:11:55 +0000
2   From: Alan MacSimoin <sujobs@tcd.ie>
3   To: SUjobs-list L-Z:  ;
4   Subject: Vacancies, 29.07.2003
5
6   OCCASIONAL WORK - MYSTERY SHOPPER
7
8   Are You:
9   .. Looking to earn additional income on a flexible basis
10  .. Over 21 years of age
11  .. Comfortable communicating on the telephone and face to face
12  .. Reliable
13
14  If the answer is yes to the above, we would be interested to
15  hear from you.
16
17  To learn more, go to
18  www.fspulse.co.uk/applyonline
19  ------------------------------------
20  Employment Service
21  Trinity College Students' Union
22  House 6, Trinity College, Dublin 2
23  Voice: 01-6081268   Fax: 01-6777957
```

The above email was sent to the author by the student union employment office. It is a service that keeps students informed about part time jobs available in Dublin. This email was marked as spam by SA because it failed to comply with three rules. Firstly, it had a malformed 'To' address. This can be seen on line 3. The mail was assigned 1.5 points for this. Secondly, SA assigned the mail 2.9 points for mentioning the words 'additional income' in the text (line 9). This seems to be rather excessive. Finally, it seems that the date is not functioning correctly on this persons email account. The email was actually received 14 hours *before* the date given in line 1. For this, SA awarded 2.8 points. This added up to 7.2 points, which is over the threshold of 5 points. Here we can see that as a result of breaking 3 SA rules, the mail has been marked as spam. There are many other SA rules which could easily lead to false positives. For example, if a subject line starts with the word 'hello', then SA will assign 2.7 points to the mail. If the subject starts with 'hello !!!' it is assigned 4.4 points (an extra 1.7 points are added for the exclamation marks). This is only 0.6 points away from being categorized as spam. There are many other rules which could result in false positives. The email is assigned 2.6 points if the body includes the words 'Dear *something*' where *something* is the receivers name. If the senders email address ends in a number, the mail is assigned 0.7 points. It seems that many of SA's false positives are the result of over-zealous rules which have not been properly thought out.

The CBDF filtering filtering method will not have the same problems. Firstly the CBDF does not have any heuristic rules that have been written by hand. While the SA filter can be thrown by an email containing a word like 'GUARANTEE', the CBDF filter should be unaffected. If a legitimate email contains such a word, its effect should be cancelled out by all the legitimate words that the mail will contain. Words that are unlikely to occur in spam emails decrease the probability of the email being spam. For this reason, the CBDF filter produces far less

false positives than SA.

6 Conclusions

It is obvious that the inflexible nature of SpamAssassin's rules leave it at a disadvantage. While the CBDF method can learn as it filters, SA does not have that option. It is possible for the user to change the scores of SA's rules, but this is a tedious exercise. While SpamAssassin is a rule-based technique, CBDF is character-based. Chaski (1998) has identified the character-based approach as being far more effective than a rule-based approach. While SA does have some advantages over CBDF (for example explaining exactly why a mail has been marked as spam), the latter method will produce better results. Most researchers view spam filtering as a text classification problem (Mehran Sahami and Horvitz, 1998), (Androutsopoulos et al., 2000). Viewing it as an authorship identification problem may help to dramatically reduce spam. If the "textual fingerprints" of the large-scale spammers can be identified, then their emails can be easily filtered out. Further research is needed to convert this CBDF filter from a stand alone filter to one that it fully integrated with an email account and which functions in real time.

References

Ian Androutsopoulos, John Koutsias, V. Chandrinos, George Paliouras, and C. Spyropoulos. 2000. An evaluation of naive bayesian anti-spam filtering. In *Workshop on Machine Learning in the New Information Age*.

Harald Baayen, Hans van Halteren, Anneke Neijt, and Fiona Tweedie. 2002. An experiment in authorship attribution. In *Journées internationales d'Analyse statistique des Donnèes Textuelles*.

Carole E. Chaski. 1998. A daubert-inspired assessment of current techniques for language-based author identification. Technical Report 1098, ILE Technical Report.

Paul Graham. 2002. A plan for spam. http://www.paulgraham.com/spam.html, August. Last verified, 7/10/2003.

Adam Kilgarriff and Raphael Salkie. 1996. Corpus similarity and homogeneity via word frequency. In *Proc. EURALEX '96, Gothenburg, Sweden*.

Mark Ludlow. 2002. Just 150 'spammers' blamed for e-mail woe. The Sunday Times, 1st December, page 3.

Niambh McCombe. 2002. Methods of author identification. B.A. (Mod) CSLL Final Year Project, TCD.

David Heckerman Mehran Sahami, Susan Dumais and Eric Horvitz. 1998. A bayesian approach to filtering junk e-mail. In *Learning fo Text Categorization - Papers from the AAAI Workshop*, pages 55–62.

Michael P. Oakes. 1998. *Statistics for Corpus Linguistics*. Edinburgh University Press.

Cormac O'Brien and Carl Vogel. 2003a. A forensic examination of a funerall elegy. unpublished manuscript; in preperation for peer review.

Cormac O'Brien and Carl Vogel. 2003b. Spam filters: Bayes vs. chi-squared; letters vs. words. In *Proceedings of the International Symposium on Information and Communication Technologies*.

Sofie Van Gijsel. 2002. A corpus-linguistic analysis of european right-wing party manifestos. Master's thesis, University of Dublin, Trinity College.