Spam Filters: Bayes vs. Chi-squared; Letters vs. Words

Cormac O'Brien*& Carl Vogel†

Abstract

We compare two statistical methods for identifying *spam* or junk electronic mail. The proliferation of spam email has made electronic filtering vitally important. The magnitude of the problem is discussed. We examine the Naive Bayesian method in relation to the 'Chi by degrees of Freedom' approach, the latter used in the field of authorship identification. Both methods produce very promising results. However, the 'Chi by degrees of Freedom' has the advantage of providing significance measures, which will help to reduce false positives. Statistics based on character-level tokenization proves more effective than word-level.

1 Introduction

Email is a very popular method of communication. In recent years this medium has become the target of abuse. Mass posting of unsolicited email messages (spam) has become increasingly prevalent. Spam is a huge problem for all email users, from the casual user, who loses time deleting all the junk mails before reading the legitimate ones, to the large companies who spend millions of euro a year trying to combat it. Many efforts have been made to combat spam, both technical and legal. Some have been effective, but as a new filter comes on the market, spammers try to beat it. We propose methods that would make it difficult for the spammers to find loopholes. This is because the user trains the filter using *their own email*. As the spammer has no idea of the content of peoples' individual emails, they have difficulty beating the filters. The two approaches that we address here are the Naive Bayesian method and the 'Chi by degrees of Freedom' ($\frac{\chi}{d.f.}$, henceforth) method. The Nave Bayesian method has been used several times before to filter email (e.g Androutsopoulos, Koutsias, Chandrinos, Paliouras, & Spyropoulos, 2000; Mehran Sahami & Horvitz, 1998) and has been very effective. The $\frac{\chi}{d.f.}$ method has, as far as we're aware, never been used in the area of email filtering. It has the advantage of providing significance measures. These significance measures could help to combat false positives, which is a massive consideration when designing email filters. In §2 we describe spam and the problems it creates. We outline methods used to combat spam in §3. In §4 we introduce the two statistical approaches that we experimented with. In §5 we will discuss the methods used in our experiments, and §6 presents the results. Finally, §7 is summarizes and suggests future directions.

2 Spam: What's the Beef?

While electronic mail has undoubtedly become a success over the last ten years, it is not without its problems. Spam has become a major irritant to most email users. Spam (junk email) is described in this paper as unsolicited bulk email. Virtually anything can be advertised in hope of achieving a sale. Spam also has the capacity to be harmful, slowing down critical servers and servides. Parents of children with email access may be concerned by a report from Internetweek.com (2002b) that pornographic images made up one in 3,418 emails in October 2002.

Spam is by no means a small problem. According to Brightmail[1], 40% of all mail received by Brightmail customers is spam. This is up from eight percent the previous year. Also, spam attacks have become more varied. Brightmail has shown that unique spam attacks have risen

*Computational Linguistics Group, Trinity College, U. of Dublin: obrience@tcd.ie

†Computational Linguistics Group & Centre for Computing and Language Studies, Trinity College, U. of Dublin: vogel@tcd.ie

[1] www.brightmail.com, last verified September 2002

from 683,579 in April 2001 to 5,285,404 in September 2002. Hotmail account holders also have trouble with spam. SPAMHAUS[2] say that spammers are carrying out a *dictionary attack* on hotmail.com. A dictionary attack is based on the fact that users of email servers like hotmail tend to decide on account names based on common words and names; but because often other people have the same choice (e.g. `ireland@_.com`) users are forced by the service provider to find a unique variant (e.g. `1re1and@_.com`) in order to avoid shared addresses. The dictionary attack simply automates this process, collecting valid guessed email addresses. Often a hotmail address can begin receiving spam within days of opening an account.

Spam is also bad news for businesses who have to pay for more bandwidth and storage space. The productivity wasted by workers who have to sift through their mail to separate legitimate mail from spam is also a factor. Internetweek.com (2002a) cites Barry Shine, president of the ISP provider 'The World' as stating that 30% of staff expenses at his 20-person company is spent either putting in filters or talking to customers on the phone about spam.

Different reasons motivate spammers. Primarily, it is a very cheap way of reaching millions of potential customers, as email it generally costs the same to send one million emails as it does to send one. This cost-effective marketing method is attractive to some. Moreover, spamming must be profitable. According to a recent article by Mangalindan (2002a) if a spammer gets as few as 100 responses for every 10,000,000 emails, they can make an attractive profit.

3 Anti-Spam Methods (I): Distrust Strangers, Hide, Sniff First, Litigate, Legislate

Spam, unchecked, has the ability to stall internet email altogether. Unless something combats spam effectively, email may become useless as a tool for communication. As spam has become a larger problem, more effort has been put into combating it. Start-up IT companies developing spam blocking methods are still able to attract venture capitalists (Mangalindan, 2002b).

One popular method to fight spammers is known as *blacklisting*. A blacklist of email addresses known to belong to spammers is set up. If the user receives an email from one of these addresses, the message is automatically marked as spam. One problem with this method is that non-spammers may have their addresses placed on the list accidently (this may lead to legal problems). Also, one feature of spammers is that they tend to use "shell addresses" (ie. an address that they use onece and then discard) when sending spam. They can change their addresses regularly to insure against being blacklisted. The opposite of this is for the user to create a *whitelist* containing all the addresses that the user deems to be legitimate, deeming all other emails as spam. This is too inflexible resulting in the loss of much non-spam emails.

An intermediate 'greylist' approach is used by*Matador* from Mail-Frontier, Inc. If the program receives an email that it suspects might be spam, it places it in a special "challenge" folder. Then the program sends the sender a message containing a random picture of animals. It asks the sender to reply saying how many animals are in the picture. If the sender replies with the correct answer, the mail is placed in the users in-box. Most spammers won't read any replies that are sent to it, so their spam mails won't get past this program. Of course this method will have certain limitations. It may take some time for legitimate emails to be read, depending on how often the sender checks their in-box and replies to the animal question. It may also eliminate legitimate email from senders who use only text-based email readers and thus cannot see the graphics.

Munging is a way for email users to ensure that spammers do not get hold of their address in the first place. Munging is when the user modifies their email address so that emails sent to the modified address won't reach the user. This is a common practice in posts to Usenet. For example, if the users name was *cormac@we_hate_spam.com* they could munge it to *cormac(at)we_hate_spam(dot)com*. Now email harvesters[3] will not recognise it as an email address

[2] www.spamhaus.org, last verified 5th Jan. 2003

[3] Spammers use programs to search public areas on the internet to compile email lists.

and the user will not receive spam as a result of posting messages on Usenet

The most popular method of email blocking is filtering (though spam is no longer easily identifiable by simple key-word search). Rule-based filters such as SpamAssassin use a wide range of heuristic tests to identify spam. Tests include searching the email for an opt-out clause. If a certain number of tests are positive, the mail is classed as spam. SpamAssassin has a problem with false positives: of the 75 mails marked spam in the first author's inbox, 32 of them are in fact legitimate. That means that 42% of the mails are marked incorrectly as spam in the inbox. Most of the mails incorrectly marked as spam are newsletters or automated replies. If the filter was trained on the personal inbox, these mails would probably not have been marked as spam.

The anti-spam company Habeas provide a litigious way to counter spam. Habeas owns the trademark and copyright on a haiku poem.[4] The company then allows users to mark their emails as "Sender Warranted Email" by placing the haiku in the header of their mail. The filters will then search the headers for the poem. If it contains the header, the mail is automatically allowed through. However, spammers should be wary of attaching the haiku to their junk mails. Haebas has said that it will pursue offenders through the courts. The legal aspects are interesting. Normally the header tag would be too small to be copyrighted, but poetry, however small is covered by copyright law. Also almost all countries have laws which protect intellectual property such as Habeas' haiku. This means that spammers will be unable to move their operations to countries with lax anti-spam legislation. How effective this method is remains to be seen, but Habeas is currently taking two bulk-mailers to court for including the haiku in their email headers.

Legislators have tried to combat spam as well. In Europe, the European Union passed the "E-Privacy Directive" (directive 2002/58/EC) concerning the processing of personal data and the protection of privacy in the electronic communications. This directive essentially made it illegal to send "opt-out" junk mail. Opt-out spammers assume that everyone wants junk mail, and the only people who don't are those who avail of the opt-out clause that can be found at the bottom of the junk mails. The E.U. directive says that all bulk email should be opt-in. That means that the people who receive the mail have stipulated that they want to receive information about the product being advertised (usually by clicking a box on a web-page). This E.U. directive has not been very effective, mainly because most of the spam, comes from outside the European Union. In the United States, 27 states have laws on their books to combat spam. However at present there is no federal laws against spam. This makes it easy for spammers to operate in states where spam is not illegal. Florida is a popular state for spammers. Until spamming is made illegal in all of the United States, spammers can operate with impunity.

4 Anti-Spam Methods (II): Learn by Tasting (Bayes vs. $\frac{\chi}{d.f.}$)

Statistical filters are quite successful. In part, this is because of the potential they have for training on the email individual users tend to receive. The two approaches used to test for spam were the Naive Bayesian method and the $\frac{\chi}{d.f.}$ method.

The Naive Bayesian method was used to filter email by both Androutsopoulos et al. (2000) and Graham (2003). In naive Bayesian classification, each email is represented by a vector $x = \langle x_1, x_2, x_3, \ldots, x_n \rangle$, where $x_1, \ldots, x_n$ are the values of attributes $X_1, \ldots, X_n$, and n is the number of attributes in the corpus of emails that has been collected. Here, each attribute represents a particular word occurring or not. If the email contains the word corresponding to x_i, the $x_i = 1$ otherwise $x_i = 0$. Androutsopoulos et al. (2000) use Bayes's theorem and the theorem of total probability. This says that given the vector $\vec{x} = \langle x_1, x_2, x_3, \ldots, x_n \rangle$ of a document d, and where where $k \in \{spam, legit\}$, the probability that d belongs to category c is as given in (1).

(1) $$P(c|\vec{x}) = \frac{P(c) \cdot P(\vec{x}|c)}{\Sigma_k P(k) \cdot P(\vec{x}|k)}$$

(2) $$P(c|\vec{x}) = \frac{P(c) \cdot \Pi P(x_i|c)}{\Sigma_k P(k) \cdot \Pi P(x_i|k)}$$

[4] The haiku is: winter into spring, brightly anticipated, like Habeas SWE$^{(TM)}$

Androutsopoulos et al. (2000) notes that the probabilities $P(\vec{X}|C)$ (ie. the probability of vector $\langle 1, 0, 0, 1, 0, 1, \ldots \rangle$ given C) are almost impossible to calculate, due to the fact that the there are too many possible values for $\vec{X}$ even though it is binary in nature. There are also data sparseness problems. Instead, the Naive Bayesian classifier makes the assumption that $X_1, \ldots, Xn$ are conditionally independent of category C. This means that we can change the above equation to the one given in (2). The difference between the two is that it is much easier to calculate $P(x_i|c)$ than to calculate $P(\vec{x}|c)$. For example, it is far less difficult to calculate $P(\text{“word”}|CategoryA)$ than to calculate $P(\langle 1, 0, 0, 1, 0, 1, \ldots \rangle|CategoryA)$. In this instance, $P(X_i|C)$ and $P(C)$ can easily be calculated using the relative frequencies from our training corpus.[5] This is a computationally efficient classifier. However, Sahami (1996) notes the strong independence assumptions are often violated in practice, leading to poor predictive generalization.

The Naive Bayesian filter calculates the likelihoods of all the words in the given email. The probability of the mail being spam is estimated using equation (2). An effective way to combat false positives is to regard as spam only mails that have a probability higher than a named threshold. Graham (2003) uses a threshold of 0.9 for his Bayesian filter, arguing that few probabilities end up in the middle of the range. The higher the threshold, the less likely are false positives. In our experiments, most spam have a probability close to 0.99 and most legitimate mails have a probability close to 0.01. Thus, mid-range thresholds will not result in false negatives.

The second approach is the $\frac{\chi}{d.f.}$ method. This method is often used in authorship identification, as they may prove to be successful in detecting spam. According to Ludlow (2002) the vast majority of the tens of millions of unsolicited emails might be the work of only 150 spammers around the world. Baayan, van Halteren, Neijt, and Tweedie (2002) argue that “authors may have textual fingerprints, at least for texts produced by writers who are not consciously charging their style of writing across texts”. If this is the case, we could use authorship identification methods to identify these textual fingerprints and eliminate a large proportion of spam.

To carry out the $\frac{\chi}{d.f.}$ 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 (2003) 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 $\frac{\chi}{d.f.}$ 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[6] 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.

A problem with using the chi-square test alone is that the larger the sample size, the more likely the null hypothesis will be rejected. McCombe (2002, pg. 31) says that “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”. For this reason, the $\frac{\chi}{d.f.}$ test is used. The programs return a table of the similarity scores of each of the corpus pairs. From this we can then deduce whether the email in question is similar to the spam corpus or to the legitimate corpus.

[5] Suppose the word “FREE” appears in a training corpus of 10,000 words 100 times, and that 87 of these are from spam, while 13 come from legitimate mails. Assuming the corpus is made up of 5,000 words of spam and 5,000 words of legitimate email, then $P(\text{“FREE”}|Spam) = \frac{87}{5000}$ or 0.02.

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

4.1 Data Collection: How to Stock up on Spam

To train the algorithms, a corpora of spam and legitimate emails had to be compiled. The corpora had to be reasonably large, so that our algorithms could estimate probabilities such as $P(\text{“FREE”}|Spam)$. Several corpora are freely available on the Internet. Examples include the Lingspam Corpus[7] which is made up of spam and legitimate mail sent to the Linguist list, and ‘The Great Spam Archive’[8] which contains 15369 spam messages at the last count. Although Androutsopoulos et al. (2000) used the LingSpam Corpus, we decided to experiment with spam we received. The reason for this, is that all users are advised to compile their own corpus when using statistical filters. By doing this, the user makes it impossible for spammers to come up with a spam mail guaranteed to pass through anti-spam filters. The spam was taken from a hotmail account that we already possessed. Our corpus contains 499 spam messages for training and 39 messages for testing. We also had 260 legitimate emails for training and 28 for testing. Some editing of the emails was done when converting them into text files. Firstly any images contained in the email was changed into the string [IMAGE]. This is because our filters are text based. Although the filters can not analyze images, it is important for them to know that they are present. In our corpus,[9] spam contain images more often. Knowledge of this should help the filters to identify spam. All header information from the emails was kept. Drucker, Wu, and Vapnik (1999) showed that better results are obtained when the header information is kept. Both algorithms were trained using the corpora and then tested on fresh testing data.

4.2 Digesting the Results: Bayes vs. $\frac{\chi}{d.f.}$; Characters vs. Words

Some very promising results were returned for both algorithms, as can be seen in table 1.A. The Bayesian algorithm is very effective when using characters as tokens, but its performance diminishes when words are used as tokens. The Bayesian algorithm using characters as tokens and the $\frac{\chi}{d.f.}$ algorithm using words as tokens both have an error rate of 0.0. This means that they assigned each email correctly, with no false positives or false negatives. Next comes the $\frac{\chi}{d.f.}$ algorithm using characters as tokens with an error rate of 0.015. Here the program had one false positive, where a legitimate mail was marked as spam. Many would argue that marking a legitimate mail as spam is far more serious than marking a spam mail as legitimate. While this is true, the $\frac{\chi}{d.f.}$ algorithm using characters is still performs better than the Bayesian algorithm using words, which has an error rate of 0.13. It marked 9 spam messages as legitimate (about 25% of the spam messages). Its recall is an unimpressive 0.769.

A: RESULTS

	Bayesian Word	Bayesian Char	Chi Word	Chi Char
Error rate	0.13	0.0	0.0	0.015
Precision	1.0	1.0	1.0	0.975
Recall	0.769	1.0	1.0	1.0

B: DEFINITIONS

Recall	Precision	Error
$\frac{TP}{TP+FN}$	$\frac{TP}{TP+FP}$	$\frac{FP+FN}{TP+FP+FN+TN}$

Table 1: Results of the experiments & Definitions of Recall, Precision and Error

The results show that both the Bayesian and $\frac{\chi}{d.f.}$ algorithms are excellent at filtering emails. The results also seem to show that the tokens chosen make a difference in the end results. TP(True Positive) is the number of spam messages marked as spam. FN(False Negative) is the number of spam messages marked as legitimate. FP(False Positive) is the number of legitimate email marked as spam. TN(True Negative) is the number of legitimate emails marked as legitimate. Table 1.B shows how we calculated the Recall, Precision and Error.

[7] http://www.iit.demokritos.gr/ ionandr/

[8] http://www.annexia.org/spam/index.msp

[9] This was constructed primarily from email sent to the first-author.

In the introduction we mentioned that the $\frac{\chi}{d.f.}$ has the advantage of providing significance measures. As we said earlier, significance is a measure of the chance that rejecting the null hypothesis is wrong. We suggested that this would help to identify false positives. We decided that if the probability of rejecting the null hypothesis being wrong was greater than 0.25, then we couldn't rely on the classification. This was not as effective as we had hoped. The $\frac{\chi}{d.f.}$ algorithm only gave us one false positive from the test data. In this instance, the probability of the null hypothesis was less than 0.25, so the algorithm didn't spot it. In most classifications, the significance level was between 0.1 and 0.25 while in one classification the significance level was less than 0.005 meaning that the program was almost certain about its classification.

5 Conclusions: hope for evegetarians

We have described application of authorship identification techniques to spam identification. We compared it to the common Bayesian method and shown that it comes up with equally as good or better results depending on the tokens selected, character rather than word-level tokenization being optimal for both methods. Others such as Graham (2003) and Androutsopoulos et al. (2000) have shown that the Naive Bayesian filter is very effective. Further research should be carried out on the $\frac{\chi}{d.f.}$ algorithm to see if it is as successful with a much larger test and training corpus. In particular we are interested to see if the use of significance levels would help to filter out false positives. It should be pointed out that the results are based on training with our own email. We intend for the approach to incorporate learning as FN emails inevitably slip through. This will have the advantage of making it very difficult for spammers to knowingly write email that can beat the filter. It remains to be seen if the $\frac{\chi}{d.f.}$ filter could be incorporated into a commercial email system, but the results suggest that it is an option.

References

Androutsopoulos, I., Koutsias, J., Chandrinos, V., Paliouras, G., & Spyropoulos, C. (2000). An Evaluation of Naive Bayesian Anti-Spam Filtering. In *Workshop on Machine Learning in the New Information Age*.

Baayan, H., van Halteren, H., Neijt, A., & Tweedie, F. (2002). An Experiment in Authorship Attribution. In *JDAT*.

Drucker, H., Wu, D., & Vapnik, V. (1999). Support vector machines for Spam categorization. *IEEE-NN*, *10*(5), 1048–1054.

Graham, P. (2003). Better Baysian Filtering. In *Proceedings of Spam Conference* http://spamconference.org/proceedings2003.html.

Internetweek.com (2002a). Spam Is A Thousand Times More Horrible Than You Can Imagine. http://www.internetweek.com/story/INW20021219S0003.

Internetweek.com (2002b). Study: e-mail viruses up. http://www.internetweek.com/story/INW20021109S0002.

Kilgarriff, A. & Salkie, R. (1996). Corpus similarity and homogeneity via word frequency. In *Proc. EURALEX '96, Gothenburg, Sweden.*

Ludlow, M. (2002). Just 150 'spammers' blamed for e-mail woe. The Sunday Times, 1st December.

Mangalindan, M. (2002a). Some Bulk E-Mailers Make a Healthy Living On Steady Diet of Spam. The Wall Street Journal Europe, 13th November.

Mangalindan, M. (2002b). Spam Busters Draw Investor Interest. The Wall Street Journal Europe, 20th June.

McCombe, N. (2002). Methods of Author Identification. B.A. (Mod) CSLL Final Year Project, TCD.

Mehran Sahami, Susan Dumais, D. H. & Horvitz, E. (1998). A Bayesian Approach to Filtering Junk E-Mail. In *Learning fo Text Categorization - Papers from the AAAI Workshop*, pp. 55–62.

Oakes, M. P. (1998). *Statistics for Corpus Linguistics*. Edinburgh University Press.

O'Brien, C. & Vogel, C. (2003). A Forensic Examination of A Funerall Elegy. unpublished manuscript.

Sahami, M. (1996). Learning Limited Dependence Bayesian Classifiers. In *KDD-96: Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, pp. 335–338.

Van Gijsel, S. (2002). A Corpus-Linguistic Analysis of European Right-Wing Party Manifestos. Master's thesis, University of Dublin, Trinity College.