Word Stemming to Enhance Spam Filtering

Shabbir Ahmed and Farzana Mithun

Department of Computer Science & Engineering, University of Dhaka, Bangladesh
{shabbir,mithun}@acm.org

Abstract. Generally a content based spam filter works on words and phrases of email text and if it finds offensive content it gives that email a numerical value (depending on the content). After crossing a certain threshold, that email may be considered as SPAM. This technique works well only if the offensive words are lexically correct. That means the words must be valid words with correct spelling. Otherwise most content based spam filters will be unable to detect offensive words. In this paper, we showed that if we use some sort of word stemming or word hashing technique that can extract the base or stem of a misspelled or modified word, the efficiency of any content based spam filter can be significantly improved. Here we presented a simple rule-based word stemming algorithm specifically designed for spam detection and showed some experimental results to corroborate our claim.

1 Introduction

Content based spam filters are useless if they cannot 'understand' the 'meaning' of the words or phrases in an email. Nowadays, spammers change one or more characters of offensive words in their spam in order to foil content based filters. But the important thing to observe is that the spammers change the words in such a way that a human being can understand the meaning the words without any difficulty. That's why one might wonder why his spam filter does not detect some emails as spam when he can clearly see that those are spam! Spammers do not make any drastic change in the words so that it can be easily recognized by humans. Based on the above mentioned observations, we developed a rule based word stemming [1] technique that can match words those both look alike and sound alike.

For example, the versions of the word 'Viagra', 'Via*gra', 'Vi\gra!', 'V.i-a.g*r.a' etc. cannot be detected by conventional spam filters.

In the section 2, we outlined our word stemming technique. Experimental setup and results were described in section 3 and finally in section 4, we concluded our topic.

2 Word Stemming

Algorithm for processing a word (stemming/hashing):
1) Remove all non-alpha characters (but allow some characters like '/' '\' '|' etc. which can be used together to look like some characters, such as \/ for 'V'). [Detailed statistics required]
2) Remove all vowels from the word except for a trailing one.
3) Replace consecutive repeated characters by a single character.
4) Use phonetic algorithms like soundex on the resultant string.
5) Give it a numeric value depending on the operations performed over it.
6) Use this resultant string (numeric value) to look up a table (that contains a list of offending words where each word has a range of acceptable values)
7) Replace original word with that of the table.

Word boundary detection is crucial in this case. Some points to consider about word boundary detection are: (a) How many words can there be in a single line (80 character line)? (b) How many delimiter characters or special characters can be found in a line? (c) Suspect two or more short consecutive words. (d) Suspect a line with many special characters, many words etc.

3 Experiments

Our experimental setup was shown in Fig. 1 (we didn't bother about filter performance or integrity of the messages; obviously a dual MTA setup with an intercepting daemon like Amavisd [2] would be a better choice). We used Sendmail [3] as the MTA and SpamAssassin [4] as the spam filter and Procmail [5] as the local delivery agent. The experimental result was shown in chart 1. Two sets of data were plotted. The data of set 1 were collected from 2933 emails from June '03 to Oct '03 and the data of set 2 were collected from 3954 email from Jan '04 to Mar '04. We can see from the charts that the efficiency of spam filters went down in recent days. This is due to the fact that spammers are increasingly using other techniques to fool Baysian filters [6].

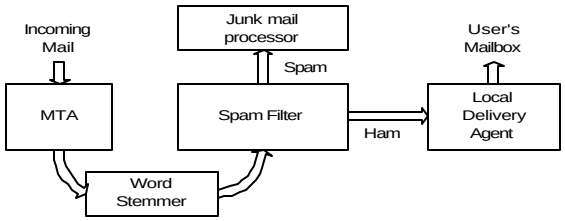

Fig. 1. Experimental setup showing the position of word stemmer

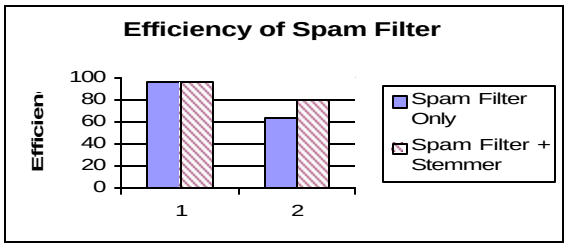

Chart 1. Performance comparison of Spam filter with word stemmer

4 Conclusions

The fact that spammers modify words in such a way so that the words can be easily recognizable by a human being was the key to build this word stemming technique. As it can be seen from section 3, our technique improves the spam detection process when used together with a traditional spam filter. Our method seamlessly integrates with existing mail tools - no modification to spam filters or MTAs is needed in order to use our technique

The algorithm presented here was simply rule-based. This type of rule-based technique is easy to defeat, once the spammers know how the word hashing works. An interesting research direction is to devise some sort of dynamic word hashing or word squeezing or word stemming algorithm that would be difficult to foil. The technique presented here was not designed with optimization in mind. In order to make it more practical one need to seek for all sort of optimization that can be performed on this technique. Word boundary detection is another problem for spam filters. This paper does not cover that topic though it gives some idea in section 2 what should be considered when devising algorithms for detecting word boundary. A good word boundary detection technique should be used in conjunction with our proposed technique in order to make it useful.

References

1. Orwant J. et al. *Mastering Algorithms with Perl*. O'Reilly and Associates, ISBN: 1-56592-398-7, 1999.
2. Amavisd-new Home Page, http://www.ijs.si/software/amavisd, Accessed 01 July 2004.
3. Sendmail Home Page, http://www.sendmail.org, Accessed 01, July 2004.
4. SpamAssassin Home Page, http://www.spamassassin.org, Accessed 01, July 2004.
5. Procmail Home Page, http://www.procmail.org, Accessed 03, Mar 2004.
6. Graham, P. *Better Baysian Filtering.* In Proceedings of Spam Conference, 2003.