The Bayesian Spam Filter with NCD*

Michal Prílepok[1], Jan Platoš[1], Václav Snášel[1], and Eyas El-Qawasmeh[2]

[1] Department of Computer Science, FEI, VSB - Technical University of Ostrava,
17. listopadu 15, 708 33, Ostrava-Poruba, Czech Republic
{michal.prilepok, jan.platos, vaclav.snasel}@vsb.cz
[2] King Saud University, Saudi Arabia
eyasa@usa.net

Abstract. Undesired e-mail (spam) becomes a big problem nowadays not only for users, but also for Internet providers. One of the main obstacles for elimination of this problem is a complicated security issues. In particular, the very low rate of falsely detected e-mails in practice. We tried to eliminate this problem by a sufficient similarity check of checked e-mails with e-mails marked as unrequested or spam. This paper uses Bayesian algorithm with many variations. For comparison purposes, we used a normalized compression distance which helped us reduce the rate of false detection of individual mails.

Keywords: e-mail, spam, Bayesian filter, data compression

1 Introduction

Spam senders are flooding us with increasing number of unrequested e-mails. Such practice makes us think about developing of defensive techniques. Currently, there are many developed techniques and approaches, which enable us to eliminate unrequested e-mails. These techniques and approaches can be divided into the following categories:

- Sender or mediator analysis
- E-mail content analysis
- Sponsor analysis

Neither of the given categories is separately effective enough. Nowadays, combination of various techniques and attitudes occurs in order to improve success of fighting against spam [10].

* This work was partially supported by the Grant Agency of the Czech Republic under grant no. P202/11/P142, SGS in VSB Technical University of Ostrava, Czech Republic, under the grant No. SP2012/58, and has been elaborated in the framework of the IT4Innovations Centre of Excellence project, reg. no. CZ.1.05/1.1.00/02.0070 supported by Operational Programme 'Research and Development for Innovations' funded by Structural Funds of the European Union and state budget of the Czech Republic.

J. Pokorný, V. Snášel, K. Richta (Eds.): Dateso 2012, pp. 60–68, ISBN 978-80-7378-171-2.

The organization of this paper will be as follows. Section 2 describes current related work. Section 3 contains description of the Bayesian spam filter. Section 4 describe normalized compression distance. Section 5 defines the interconnection of the Bayesian Spam filter an NCD. The results of our experiments are described in Section 6 and Section 7 contains conclusion of this paper.

2 Current Related Work

First techniques of fighting against spam came out of detection key words in the e-mail subject. These spam filters compared words contained in e-mail subject with the list of prohibited words. Spam authors started to avoid this way of spam detection by several modifications of e-mail subjects. Modification of e-mail subject is often used and makes us think that it was a forwarded e-mail from the third person, e.g. "RE: About us" [6, 10].

Later, filter techniques started to use body of the e-mail as well. In the same way, words from the e-mail body were compared to the list of prohibited words. This extension brought only a little improvement. Rate of success in detection of unrequested e-mail was dependent on the quality of the list of prohibited words. Individual prohibited words had to be chosen very carefully so that increasing in final false detection rate would not happen.

Filters use check-sums and marks for elimination of false detection rate. The check-sums are calculated over each received e-mail and resulting hash are compared with database of known spam e-mails. One-way transformation function also called hash function (MD5, SHA1, ...) are used as check-sum calculators.

Spam filtering based on key words [10] did not bring required success of unrequested e-mails detection. Analysis of recent e-mails trend seems to be a very effective method in fighting against unrequested mail. This technique reduces the rate of unrequested e-mail false detection. Heuristic filters are ranked to these techniques - rules-based filters and learning-based filters.

Rules-based filters [10] are looking for features which are characteristic for spam in the e-mails. These are some words (e.g. viagra), collocations and mistakes typical for spam. Example of such mistake can be sending e-mail with a future date, prohibited marks in the heading, incorrectly marked MIME type of e-mail, etc. Each detected feature has defined certain points score. Usually, points are summed up and if the sum exceed defined limit, the email is marked as a spam. Detected features are defined by the help of rules that have to be regularly updated and conformed to spamers practices.

Learning-based filters (often called bayesian) [7, 4, 2, 1] use tricks from the area of artificial intelligence. In the learning phase, the email is submitted into filter. Each email is marked as spam of ham (not spam). Filter extracts features from each email and stores them into database. Usually, the e-mail divided into words (eventually other text segments) and a probability of individual words is computed and statistically evaluated. The words probability for spam and ham emails is evaluated.

In detection phase, the filter uses collected information and computed the probability that the tested email is spam. The most often formula for probability calculation was suggested by mathematician Bayes. Learning filters are the most efficient if they are taught what is and what is not spam by end users according to their individual opinion. The Bayesian filters are also used in servers where the learning is made by all users together.

Even the Bayesian filter shows a high success of unrequested e-mail detection, false marking or non-detection still happens in some cases. This imperfection can be eliminated by comparison of the examined e-mail content with unrequested e-mails known in advance [6, 10].

3 The Bayesian spam filter

Bayesian spam filter is a statistic technique for filtering e-mails. It uses naive Bayesian classifiers for spam identification. The Bayesian classifiers work with relations between elements (typically words) from unrequested (spam) and requested e-mails. They calculate probability whether an e-mail is spam or not by the help of the Bayesian statistics. Particular words have a particular probabilities of occurrence in unrequested e-mails and legitimate e-mails. Filter does not recognize these probabilities in advance. It first has to learn them so that he could build upon them. Each new e-mail must be manually marked whether it is spam or is not. For all words in each e-mail the filter must adjust probability with which the given word occurs in spam or in legitimate e-mails in its database. For instance words ”viagra” or ”refinance” are often found in spam e-mails and names of friends or family members are often found in legitimate e-mails.

3.1 Calculation of probability that e-mail containing a word is spam

Probability of e-mail which contains a particular word can be calculated by the help of the following formula [7, 4, 2, 1]:

$$\Pr(S \mid W) = \frac{\Pr(W \mid S) \times \Pr(S)}{\Pr(W \mid S) \times \Pr(S) + \Pr(W \mid H) \times \Pr(H)}$$

where:

- $\Pr(S \mid W)$ is probability that e-mail is spam with knowledge that it contains the examined word.
- $\Pr(S)$ is total probability that e-mail is spam.
- $\Pr(W \mid S)$ is probability that the examined word occurs in spam
- $\Pr(H)$ is probability that the given e-mail is not spam (ham)
- $\Pr(W \mid H)$ is probability that the examined word occurs in ham e-mail

Probabilities for $\Pr(W \mid S)$ and $\Pr(W \mid H)$ will be determined in learning phase of the filter. By the help of $\Pr(H)$ and $\Pr(S)$, it may potentially affect partiality or impartiality of the filter against the checked mails. The more is value of $\Pr(S)$ closer to 1.0, the more is filter partial against spam mails. The value $\Pr(H)$ adjust the filter's opposite. The more is this value higher, the less is the filter partial against spam mails. Summary of values $\Pr(H)$ and $\Pr(S)$ must be equal to 1.0. The statistics presents that the probability of spam is approximately 80%. On the basis of this statement we can determine values for Pr(s) = 0.8, Pr(h) = 0.2. In this case, the Bayesian filter anticipates that 80% of checked e-mails are spams and remaining 20% are legitimate ham mails. The majority of the Bayesian filters for detection uses a hypothesis that incoming e-mails contain less spam than legitimate e-mails (ham). Therefore, they have adjusted both probabilities to 50% ($\Pr(S) = 0.5$; $\Pr(H) = 0.5$)).

It can be said about the filters which use this hypothesis that they are impartial, they do not have any prejudice against incoming mails. This hypothesis enables simplification of the general formula to:

$$\Pr(S \mid W) = \frac{\Pr(W \mid S)}{\Pr(W \mid S) \times \Pr(W \mid H)}$$

This number is called *spamcity* or *spaminess* of the examined word. The value of $\Pr(W \mid S)$ that is used in this formula is rounded to frequency of e-mails containing the examined word in e-mails marked as spam during the learning phase. Similarly, the $\Pr(W \mid H)$ is rounded to frequency of e-mails containing the examined word in e-mails marked as ham during the learning phase. The collection of e-mails determined for learning has to be representative enough due to these approximations. Data files of ham and spam e-mails should be in accordance with a 50% hypothesis of the same size. Determination whether the e-mail is spam or ham, just on the basis of a single word, is prone to mistake. That is why the Bayesian filter tries to take into account several words and interconnect their spamcity in order to determine total probability.

3.2 Combination of individual probabilities

The Bayesian filter of unrequested e-mail assumes that words are independent. This is bad in natural languages where probability of detection of an adjective is affected, e.g. by probability of a noun. Considering this assumption we can deduce further formulas of the Bayesian theorem:

$$p = \frac{p_1 \times p_2 \ldots p_n}{p_1 \times p_2 \ldots p_n + (1 - p_1) \times (1 - p_2) \ldots (1 - p_n)}$$

where p is probability that the suspected e-mail is spam and p_i is probability that $\Pr(S \mid W_i)$ is spam containing the $i - th$ examined word.

The result p is compared to a specific value. If the result p is higher than the given limit, then the email is considered as a spam, otherwise it is a ham mail.

3.3 Use of rare words

In case that the word does not occur during the learning phase, numerator and denominator is equal to zero in general, but also in spamcity formula. Software may leave out the words which do not provide any information. Words that occurred several times only in the phase of learning may cause a problem because it would be a mistake to believe to a blind information. Solution of this problem is to prevent of acceptance of such words into account. The Bayesian theorem is applied several times, and the division between spam and ham e-mails containing the examined word is a random quantity with beta distribution. Therefore, we may use a modified formula for calculation of probability:

$$\bar{\Pr}(S \mid W) = \frac{s \times \Pr(S) + n \times \Pr(S \mid W)}{s + n}$$

where

- $\bar{\Pr}(S \mid W)$ is corrected probability of spam e-mail with knowledge that it contains the examined word.
- s is a strength by which we give basic information about incoming unrequested mail.
- $\Pr(S)$ is probability that incoming e-mails are spams.
- n is a number of occurrences of the examined word in the course of the learning phase.
- $\Pr(S \mid W)$ is spamcity of the examined word.

This corrected probability is used instead of spamcity in combined formula. $\Pr(S)$ may equal to 0.5 in order to avoid too big distrust towards incoming mails. Three is a good value for s, which means that the examined word must exist three times within learning e-mails in order to increase trust of spamcity value as a default value. This formula may be extended in case when n is equal to 0 (and where spamcity is not defined). In this case, $\Pr(S)$ is evaluated.

3.4 Other heuristics

Neutral words like *the*, *and*, *some*, or *is* (in English), or their equivalents in other languages may be ignored. Generally said, some Bayesian filters ignore all words which has spamcity around 0.5, because they bring insufficiently good decision. Words taken into account should have spamcity close to 0.0 (distinctive mark of legitimate mail), or 1.0 (distinctive mark of spam mail). Method may look like this for example: 10 words that have the highest absolute value $|0.5 - p|$.

Some software products consider the fact that the given word occurs several times in the examined e-mail and others not.

Some software products use samples (word sequences) instead of separated words of natural language. For instance, they calculate the spamcity value of four words ”Viagra is good for”, instead of calculation of spamcity values for each word ”Viagra”, ”is”, ”good” and ”for”. This method provides a higher sensitivity to context and leads to better elimination of the Bayesian noise to the detriment of a bigger database.

4 Normalized Compression Distance

Normalized Compression Distance (NCD) is a mathematical way for measuring similarity of objects. Measuring of similarity is realized by the help of compression where repeating parts are suppressed by compression. It is based on algorithmic difficulty of the Normalized Information Distance (NID) developed by Andrey Kolmogorov. NCD may be used for comparison of different objects, such as music, texts or gene sequences. We may use NCD for detection of plagiarism and visual data extraction [9].

Resulting rate of probability distance is calculated by the following formula:

$$NCD = \frac{C(xy) - \min(C(x), C(y))}{\max(C(x), C(y))}$$

Where:

- $C(x)$ is size of compressed file x.
- $C(y)$ is size of compressed file y.
- $C(xy)$ is size of compressed file created by interconnected files x and y.
- $\min\{x, y\}$ is minimum of values x and y.
- $\max\{x, y\}$ is maximum of values x and y.

The NCD is in the interval $0 \leq NCD(x, y) \leq 1$. If $NCD(x, y) = 0$, then files x and y are equal. They have the highest difference when the result value of $NCD(x, y) = 1$.

4.1 Implementation

In our approach we use a GZIP program for data compression. GZIP internally use a DEFLATE compression algorithm [3]. Deflate algorithm is based on the variant of LZ77 algorithm [11] called LZSS [8]. It also uses a semi-adaptive version of Huffman encoding [5]. LZ77 algorithm and its variants belong to the dictionary based compression algorithms which replace a symbol (bytes, characters, etc.) sequences by references into dictionary. LZSS algorithm uses two types of reference. The first type of reference in not reference at all because it represents one symbol. The second type of reference is a position and length of the same sequence in the already encoded text. All three parts - characters, positions and lengths are encoded by Huffman encoding, but each element has its own model, which increases the compression efficiency.

This algorithm is widely used in data compression. Because of popularity of this algorithm and its simplicity, it is very good choice for our purpose because its implementation is very efficient and fast. Therefore, involvment of NCD into spam detection will not reduce the speed of the decision engine.

5 Interconnection of the Bayesian spam filter and NCD

We may complete the Bayesian spam filter with further helping techniques that enable us to increase detection probability of unrequested email messages. One of the possibilities is comparison of the checked e-mail with a group of spam e-mails which were used for learning of individual probabilities of the Bayesian filter.

Additional check is applied in e-mails where spamcity is higher than 0.5. For each e-mail whose spamcity value is higher than 0.5, NCD-value of similarity, to the file which has approximately the same size after compression as the checked mail, is calculated.

Email with a similar size after compression is searched in the collection spam e-mails. We select the email according its size only. We take the email with the same or the size closest to the tested one. The size criterion helps in location of the possible similar one because messages with a similar size should be more similar to the checked email. Moreover, we may use a binary search algorithm for location of these emails. This step saves a lot of time and similarity values for files that have a higher or smaller size, will not be calculated. There is a small probability that files of a different size will have a similar content like the tested mail.

NCD results are values in the interval of 0 to 1, as was mentioned above. Whereas 0 stands for a maximum similarity of the tested files. In order to combine spamcity values from the Bayesian filter and NCD, it is necessary to modify NCD value by the help of the following simple formula:

$$p_{NCD} = 1 - (NCD)$$

By the help of this modification we can get probability with the same meaning like in the Bayesian filter. The more the files are similar, the closer is the value of p_{NCD} to 1. To get resulting spamcity value we have to combine probabilities from the Bayesian filter and NCD into one probability.

The combination proceeds by the help of the following Bayesian theorem:

$$P = \frac{p_B \times p_{NCD}}{p_B \times p_{NCD} + (1 - p_B) \times (1 - p_{NCD})}$$

where P is resulting probability, p_B is probability from Bayesian filter, and p_{NCD} is probability from NCD.

6 Results of unrequested e-mail detection

The Bayesian algorithm was implemented with many variations. These algorithms were tested in database of 270 045 e-mails, where 170 750 (63,23%) emails were marked as spam and 99 295 (36,77%) emails were marked as ham, i.e. legitimate mails. Test database comes from The Text REtrieval Conference (TREC) organized in years 2005, 2006 and 2007, co-sponsored by the National Institute of Standards and Technology (NIST) and U.S. Department of Defense. In our experiments we tested three versions of the algorithm:

1. Classic Bayesian filter without any modification.
2. Classic Bayesian filter with NCD, all e-mails which reached spamcity higher than 0.5 were checked by means of NCD.
3. Classic Bayesian filter with NCD, all e-mails which had spamcity interval in the range of 0.5 to 0.75 were checked by means of NCD

The results are depicted in Table 1. The Classic Bayesian filter successfully indicated 162 894 (94.50%) spam e-mails, 7 856 (4.60%) spam e-mails were not recognized. The filter incorrectly marked as spam e-mails 9743 (9.81%) of 99 295 ham e-mails. Total number of incorrectly marked e-mails was 17 599 (6.52%). Average speed of e-mail checking process was 288 e-mails per second.

The Bayesian filter combined with NCD (with spamcity¿0.5) was able to successfully identify 169 886 (99.49%) spam e-mails. There were 864 (0,51%) of unidentified spam e-mails. Number of ham e-mails that were marked as spam was 12 575 (12.66%) emails. Total error rate of this algorithm modification was 13 439 (4.98%) of incorrectly marked e-mails. Average speed of e-mails checking process was 32.83 e-mails per second.

Last modification of the Bayesian filter in which additional testing by the help of NDC was limited on spamcity range from 0.5 to 0.75, successfully identified 169 886 (99.49%) of spam e-mails. Number of unidentified spam e-mails was 864 (0.51%). In examination of (legitimate) e-mails 12 852 (12.67%) of incorrectly marked e-mails was found out. Total error rate of the algorithm was 13 446 (4.98%) of incorrectly marked e-mails. Average speed of e-mails checking was reached in the level of 192 e-mails per second.

In comparison with the Classic Bayesian filter, versions completed with NCD show a higher efficiency in detection of spam e-mails. With the higher success rate of spam detection, the rate of incorrectly marked legitimate e-mails also increased. The Bayesian filter without NCD shows a very good filter permeability in the level of 288 e-mails per second. The version with NCD is slowed down by additional check of e-mails to 32.83 e-mails per second with NCD check, where spamcity is higher than 0.5 and 192 e-mails with restriction spamcity value in the interval of 0.5 to 0.75.

The difference in effectiveness and error rate of both Bayesian filter versions completed with NCD is not high. The only difference is in the speed of filtering process, where the version with limited usage of NCD was faster.

7 Conclusions

In this paper, a novel variant of Classic Bayesian filter with combination of Normaliced Compressed Distance was described. This combined filter was tested as filter for spam identification. In addition to Classical implementation of Bayesian filter, two versions of combination with NCD were implemented. The first version uses NCD for all emails which have spamcity higher than 0.5. The second version

Table 1. Results of the three tested algorithms

		Bayesian filter	Bayesian filter	Bayesian filter
			$NCD > 0.5$	$0.5 > NCD < 0.75$
Spam	Success rate	95.40%	99.49%	99.49%
	Error rate	4.60%	0.51%	0.51%
Ham	Success rate	90.19%	87.64%	87.33%
	Error rate	9.81%	12.66%	12.67%
Total error rate		6.52%	4.98%	4.98%
Filter speed		288.36	32.93	192.59

uses NCD only, when the spamcity was in the interval from 0.5 to 0.75. Both filters have the same efficiency in detection of spam emails which was 99.49%. The second version is much faster than the first version and its speed is almost the same as speed of Classical Bayesian filter. Both new developed versions have worse efficiency in successfull marking of non spam emails. The overall efficiency of both new algorithm was better than the original filter.

References

1. T. Almeida, A. Yamakami, and J. Almeida. Evaluation of approaches for dimensionality reduction applied with naive bayes anti-spam filters. In *Machine Learning and Applications, 2009. ICMLA '09. International Conference on*, pages 517 –522, dec. 2009.
2. Y. Begriche and A. Serhrouchni. Bayesian statistical analysis for spams. In *Local Computer Networks (LCN), 2010 IEEE 35th Conference on*, pages 989 –992, oct. 2010.
3. P. Deutsch. DEFLATE Compressed Data Format Specification version 1.3. RFC 1951 (Informational), May 1996.
4. B. C. Dhinakaran, D. Nagamalai, and J.-K. Lee. Bayesian approach based comment spam defending tool. In *Proceedings of the 3rd International Conference and Workshops on Advances in Information Security and Assurance*, ISA '09, pages 578–587, Berlin, Heidelberg, 2009. Springer-Verlag.
5. D. A. Huffman. A method for the construction of minimum-redundancy codes. *Proceedings of IRE*, 40(9):1098–1101, 1952.
6. A. Khorsi. An overview of content-based spam filtering techniques. *Informatica (Slovenia)*, 31(3):269–277, 2007.
7. Y. Song, A. Kolcz, and C. L. Giles. Better naive bayes classification for high-precision spam detection. *Softw. Pract. Exper.*, 39:1003–1024, August 2009.
8. J. A. Storer and T. G. Szymanski. Data compression via textual substitution. *Journal of the ACM*, 26(10/82):928–951, 1982.
9. P. M. B. Vitányi. Universal similarity. *CoRR*, abs/cs/0504089:5, 2005.
10. P. Wolfe, C. Scott, and M. Erwin. *Anti-Spam Tool Kit.* McGraw-Hill Osborne Media, March 2004.
11. J. Ziv and A. Lempel. A universal algorithm for sequential data compression. *IEEE Transactions on Information Theory*, IT-23(3):337–343, 1977.