Adaptive Naïve Bayesian Anti-Spam Engine

Wojciech P. Gajewski

***Abstract*—**The problem of spam has been seriously troubling the Internet community during the last few years and currently reached an alarming scale. Observations made at CERN (European Organization for Nuclear Research located in Geneva, Switzerland) show that spam mails can constitute up to 75% of daily SMTP traffic. A naïve Bayesian classifier based on a *Bag of Words* representation of an email is widely used to stop this unwanted flood as it combines good performance with simplicity of the training and classification processes. However, facing the constantly changing patterns of spam, it is necessary to assure online adaptability of the classifier.

This work proposes combining such a classifier with another NBC (naïve Bayesian classifier) based on pairs of adjacent words. Only the latter will be retrained with examples of spam reported by users. Tests are performed on considerable sets of mails both from public spam archives and CERN mailboxes. They suggest that this architecture can increase spam recall without affecting the classifier precision as it happens when only the NBC based on single words is retrained.

A reevaluation of algorithm's implementation and performance is effectuated from the perspective of over a year.

***Keywords*—**Text classification, naïve Bayesian classification, spam, email

I. Introduction

THE problem of *Spam* (Unsolicited commercial email) has become very serious for the Internet community in the last few years.

In December 2004 only less than 15% of mails arriving at SMTP gateways of CERN were legitimate (Source: statistics of CERN Mail Service ([12]). Similar information is heard from all parts of the world and one can hardly find an email user who is not familiar with this situation from his or her personal experience. This phenomenon is regarded as a threat to email user productivity, raises seriously the TCO (Total Cost of Ownership) of mail servers and finally even becomes a security issue to many organizations. From an email user perspective, it highly reduces the usability of email in everyday life.

In such circumstances it is understandable that the world of science has seen many works related to spam in recent years. From the theoretical point of view, spam fighting is a *text categorization* task and a part of the *Natural Language Processing* (NLP) field.

This paper proposes using two independent naïve Bayesian classifiers to detect spam. The first one represents an email as a vector of single words (unigrams) and its training set is constant whereas the other represents an email as a vector of pairs of adjacent words (bigrams) and its training set is supplemented by spam messages reported by users. Such a configuration results in adaptability of anti-spam engine and the improvement of classification performance.

In this section, some specific features of spam detection as a text categorization will be mentioned and the construction of so called Graham's version of naïve Bayesian classifier will be shortly discussed. The reader can also find here some comments on related work. Next, the interesting features of used algorithm will be described and test results will be presented. Finally, a short discussion will conclude the paper.

A. Spam fighting as a text classification task

At the first glance, spam detection might appear a relatively easy task for two reasons. First of all, there are only two categories while a standard text classification problem can have tens or more of them. Secondly, a definition of Spam as an unsolicited commercial mail may itself appear very hermetic and strictly related to commerce and consumption goods or some very well defined sectors of activity as pharmacy or pornographic industry. Both of them might imply that we could quickly come out with classifiers which performance is sufficient enough to eliminate the problem.

Unfortunately, even if this might have been true a few years ago, now it is not the case anymore. Spammers are aware of general methods used in anti-spam systems and take steps to overcome imposed barriers. Some messages contain a set of random words that can influence a classifier's decision. In others, the key words are divided by spaces, which still make it readable by humans, but impossible for classifiers to make a right decision. In my opinion, we could be even observing a classical 'armaments race' where an improvement on one side is immediately balanced by a reply from the other which will be again overcome by first side and so on. For extended discussion of this matter also in context of spam detection see [10].

Furthermore, most spam messages are nowadays constructed in a way that makes them as similar to legitimate emails as possible. They can resemble a message with Christmas wishes from a friend or contain only one not suspicious sentence (e.g. *Do you want a Watch?*).

These examples are proofs that the definition of spam category is fuzzy. Of course this can be equally said about any other category which will have a certain amount of examples sufficiently close to the border of category, making it

Manuscript received May 17, 2006.

Wojciech P. Gajewski is currently a Fellow in the Accelerators & Beams Department of the European Organization for Nuclear Research, Geneva, Switzerland. (email: wojciech.gajewski@gmail.com).

particularly difficult to give a confident classification. In case of spam, however, the fact that creators of those messages are purposely trying to make them indistinguishable from legitimates is becoming a serious problem.

All these factors make the spam detection task more difficult than a classical task of text classification. In the latter case, nobody is purposely trying to make both possible categories as similar as possible; neither takes steps to make the classification more difficult. On the other hand, one could find other features of spam that could ease spam detection: spam is usually sent in large series to mass recipient.

That means that when receiving a particular unsolicited email there is a very high probability of receiving other similar or identical emails at the scope of organization. Presumably, this mechanism is related to methods of updating recipient lists used by spammers.

Finally, an observation could be made that spam is changing on a daily basis. Certain mailing campaigns are still ongoing but others will quickly replace them. To deeper analyze the last observation I proposed the following experiment. From spams reported by users at CERN during a period of October 2004 I extracted all URLs. This set was then made available to SpamKiller (more information on SpamKiller and CERN anti-spam architecture can be found in [13]) system that counted a number of emails incoming to CERN containing only URLs already noted before. During the time of the experiment, no new URLs were added to the set. Daily statistics for the end of October are presented at Figure 1.

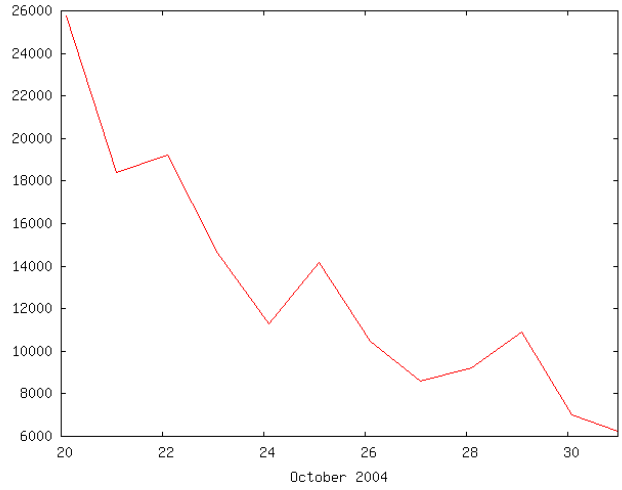

Fig. 1 Number of emails containing URLs extracted from a set of spams

We can see that after 10 days of experiment only up to 20% of the mails were not changed. Of course, a change of the URL is possibly the easiest possible to execute but this gives a clear idea how dynamic is a phenomenon of spam.

In view of these arguments it is important to ensure certain features of every spam detector. It should be:

- **Adaptive** Spam patterns are currently evolving in a very dynamic way. Anti-spam software that was pretty successful not a long time ago could today have a performance below the level of acceptability or simply be useless.
- **Online** An ideal classifier will instantly adapt itself to eliminate incorrectly classified samples.
- **Using users' feedback** Only close cooperation with mail users can provide sufficient information to allow online retraining of spam classifiers.
- **Automatic** As large as possible independence from a need of human's intervention can help to reduce maintenance costs.

Many anti-spam engines use naïve Bayesian classifier based on a single word representation of an email because of its good performance and simplicity. Online update of its spam training set can, however, deteriorate the precision of classification. I propose therefore to combine such a classifier with another naïve Bayesian classifier representing an email using pairs of adjacent words.

The latter classifier's training set of spam will be updated with new spam reported by email users. This solution, what is shown by the experiments, answers all of the mentioned needs.

B. Graham's version of naïve Bayesian classifier

Many works and practical implementations (e.g. SpamGuru, see [4]) base on the idea of *naïve Bayesian classifier* and *Bag of Words* (BOW) representation of emails to detect spam. A particular version of such classifier, different from 'classical' (described for example in [9]), was presented by Paul Graham in [1].

In general, a classification done by naïve Bayesian classifier can be described as choosing this particular category ***c*** from the set of categories *C* which is the most probable given a feature vector $\vec{w} = \begin{bmatrix} w_1 \\ \vdots \\ w_n \end{bmatrix}$, constructed of all the words found in analyzed message *m*:

$$c = \arg\max_{c \in C} P(c \mid \vec{w}) \tag{1}$$

$P(c \mid \vec{w})$ is computed using *Bayes' formula:*

$$P(c \mid \vec{w}) = \frac{P(c) \cdot P(\bigcap_{i=1}^{n} w_i \mid c)}{\sum_{k=1}^{\|C\|} P(\bigcap_{i=1}^{n} w_i \mid c_k) \cdot P(c_k)} \tag{2}$$

Naivety of the model comes from the assumption of *word independence within categories*. Therefore:

$$P(\bigcap_{i=1}^{n} w_i \mid c) = \prod_{i=1}^{n} P(w_i \mid c) \tag{3}$$

In spam detection task, a set of categories is reduced only to two elements ($C = \{s, l\}$) as we can consider an email as either being a *spam* or *legitimate* one.

In Paul Graham's classifier only the probability of

belonging to *spam* category given the feature vector $\vec{w}$ is estimated ($P(s \mid \vec{w})$).

Equation (3) applied together with *Bayes' formula* yields:

$$P(\bigcap_{i=1}^{n} w_i \mid s) = \prod_{i=1}^{n} P(s \mid w_i) \cdot \frac{P(w_i)}{P(s)} \quad (4)$$

$$P(\bigcap_{i=1}^{n} w_i \mid l) = \prod_{i=1}^{n} P(l \mid w_i) \cdot \frac{P(w_i)}{1 - P(s)} \quad (5)$$

Equations (4) and (5) applied to (2) yield:

$$P(s \mid \vec{w}) = \frac{\prod_{i=1}^{n} P(s \mid w_i)}{\prod_{i=1}^{n} P(s \mid w_i) + \left(\frac{P(s)}{1-P(s)}\right)^{n-1} \prod_{i=1}^{n} \left(1 - P(s \mid w_i)\right)} \quad (6)$$

Given a number of messages in training sets of spam and legitimate messages (N_s and N_l respectively) and number of occurrences of a word w_i in training sets (n_s and n_l) a following estimate is computed for each word during training process:

$$P(s \mid w_i) = \frac{\frac{n_s}{N_s}}{\frac{n_s}{N_s} + \frac{n_l}{N_l}} \quad (7)$$

Finally, I assume *a priori* that $P(s) = \frac{1}{2}$. Graham's version of the classifier is better suited for the spam detection task than the 'classical' one as it includes information about both categories in feature probability computing. This in turn simplifies feature vector reduction to improve classification performance.

More on 'classical' naïve Bayesian classifiers can be found in [5, 9]. To follow a detailed discussion about Graham's naïve Bayesian probabilistic techniques consult [14].

C. Previous work

There have been several attempts to use additional knowledge related to bigrams in the naïve Bayesian classification. In [7] the authors use bigrams together with single words (unigrams) as features in the NBC in the general text classification task. They note a rise of a number of correctly classified positive documents (spams). However, more negative documents were also classified incorrectly.

A comparison between a naïve Bayesian classifier based on terms and bigrams in spam recognition can be found in [8]. There, for some parameters of classifiers, the one based on bigrams can have better overall performance. However, the methodology of training process is unclear.

II. ALGORITHM

Two modules based on Graham's version of naïve Bayesian classifier [1] were investigated: one representing an email in the form of Bag of Words (using single words - unigrams) and the other representing an email as a set of pairs of adjacent words (bigrams). Between the classifiers there are also two other important differences:

The NBC based on single words generates the probability only on 15 features for which the following value is highest:

$$\left|P(s \mid w_i) - 0.5\right| \quad (8)$$

as it was described in the original Graham's algorithm. In the case of the latter NBC the same function is used to choose features (pairs of adjacent words) for the classification process. However, their number is not constant and is proportional to the message body length. It can be described as:

$$n(l(m)) = \min\left(l(m), \max(15, \frac{l(m)}{5})\right) \quad (9)$$

where *l(m)* is the number of words in the email body.

In the case of the NBC based on bigrams the assumption has been made that each feature located in a particular email and not recorded in the classifier dictionary (not found in the training set) has *a priori* low probability. Arbitrarily a value of 0.03 was chosen.

The assumption of 'default' low probability for previously unseen features in case of NBC based on bigrams is constructed as an answer to the way the classifier will be used. The training set of spams will potentially have a larger cardinality than this of solicited mails and the process of retraining will only imply enlargement of the first set. Therefore, treating bigrams unseen in the training set as belonging to 'solicited' category will result in categorizing as spam only emails very similar to examples existing in the spam training set. Very large dictionary of pair of words is encouraging us to make this assumption.

To limit dictionary size, a simple feature selection algorithm was implemented in both versions of algorithms. It deleted all words or pairs of words which occurred less than 5 times in the training sets and is based on Zipf's Law [6]. According to it we can assume that very rare words would have little informative strength and could therefore be removed from training process without much harm to classification performance.

A. Mail parsing scheme

Much care and attention was directed to assure the proper parsing of an email message. First of all, a multipart message was unified into a single mail body consisting only of textual (plain text and HTML) pieces. No part of message header is used in the classification process.

Also I ensure that transfer encoding like `Quoted-printable' and `Base64' (see [11]) are decoded as they are very often used to obfuscate emails.

In the next step HTML parsing is used to locate those parts of the text, which are invisible or very difficult to see by human reading the email. A popular spammers' technique is based on including some letters, words or even fragments of text inside the physical message with either font color equal to font background or minimal font size. This results in results in

hiding parts of a text from the eye of a user but potentially not from a classifying engine.

Finally, all the HTML tags or pseudo-HTML tags (again used by spammers to obfuscate an email) together with HTML comments and other special characters (like ASCII *Line Feed* and *Carriage Return*) are removed from mail body leaving only a sequence of length *n* of plain words. It is used to create a feature vector.

III. Empirical results

In this section the conditions of experiments and their results will be presented.

A. Training sets

Both of classifiers were taught with a basic set of *28564* solicited mails and *29074* spams. Solicited mails were randomly chosen from user mailboxes and carefully verified not to contain any spams. Spam set was collected during the whole year 2004. Those sets will be referenced to as *basic training sets* further on.

Training set of spams was enlarged by *19631* spam mails reported by CERN mail users during December 2004 and beginning of January 2005. This group of spams will be later referenced to as *additional spam set*. Both classifiers were trained with basic training set enhanced by an additional one.

B. Tests

Tests involve verifying results provided by both classifiers on a given set of spam or solicited mails. I verify how spam detectability could be improved by classifying an email as spam if any of the classifiers reports it as such. The most popular performance measures used in the literature on spam recognition are *spam precision* (*p*) and *spam recall*(*r*):

$$p = \frac{n_{s\to s}}{n_{s\to s} + n_{l\to s}} \tag{9}$$

$$r = \frac{n_{s\to s}}{n_{s\to s} + n_{s\to l}} \tag{10}$$

where $n_{s\to s}$ means number of spams classified correctly. $n_{l\to s}$ and $n_{s\to l}$ mean respectively a number of legitimate emails and spams classified incorrectly.

To facilitate presentation of results, an abbreviation NBC_1 will mean a *naïve Bayesian classifier* based on single words and NBC_2 a *naïve Bayesian classifier* based on adjacent pairs of words.

1) Spams reported by users

This test is performed on group of *39297* spam emails reported by users between December 2004 and beginning of January 2005. The set used for this test and the *additional spam set* were formed independently. Results of the test can be seen in Table I.

TABLE I
SPAM REPORTED BY CERN USERS

Classification condition	$n_{s\to s}$	Recall
NBC_1	31147	79.3%
NBC_2	24026	61.1%
$NBC_1 \vee NBC_2$	33902	86.3%
$NBC_1 \wedge NBC_2$	21271	54.1%

We can see that combining both classifiers can improve the recall by about 7%. Additional mails detected with help of NBC_2 represent nearly 34% of all of the *false negatives* not detected by NBC_1.

2) Solicited mails extracted from users

Spam precision is much more important than spam recall from the point of view of an email user who should be confident about receiving solicited mails. Solutions leading to spam recall increase should therefore be initiated with caution bearing in mind a possible decline of classification precision. This can be more costly from a user point of view than a rise of recall and therefore question the whole modification.

To analyze this problem I extracted *87730* random mails from CERN mailboxes. This group of mails was then classified with both NBCs trained with all available training sets (*basic* and *additional*). All incorrectly classified examples were verified to exclude the possibility of meeting real spam in the mails taken from user accounts.

To verify the impact on each classifier's precision, the same set of mails was then checked by NBC_1 taught only with *basic training set*.

The results are presented in Table II.

TABLE II
LEGITIMATE EMAILS

Classifier	$n_{l\to s}$	$n_{l\to s}$ after verification
NBC_1	1465	1369
NBC_2	99	4
NBC_1[a]	759	643

[a]Retrained using only basic training set

The results clearly indicate superiority of NBC_2 over NBC_1 in terms of precision. Furthermore, they show that expanding the spam training set in the case of NBC_1 can lead to serious deterioration of classifier precision. In our case we receive twice more of *false-positives*.

3) Spam from spamarchive.org

This test was performed on *5448* spams from the beginning of January 2005 stored in SpamArchive (http://www.spamarchive.org). Both NBCs were trained with *basic* and *additional* training sets. The results are presented in Table III.

TABLE III
SPAM FROM *SPAMARCHIVE*

Classification condition	$n_{s \to l}$	Recall
NBC_1	3140	57.6%
NBC_2	1979	36.3%
$NBC_1 \vee NBC_2$	3372	61.9%
$NBC_1 \wedge NBC_2$	1742	32%

The results clearly indicate that using NBC_2 alone will result in very poor recall of spam. However, combining output from both classifiers can result in detecting over 4% more of spam. Moreover, a considerable group of spam is detected by both classifiers. This fact can be used to treat those spams with a special confidence. For example they could be rejected at the SMTP level without a real risk of not delivering a real message.

A generally lower performance of NBCs than in the case of spams collected at CERN can be explained by a choice of training sets. They consist only of spams from CERN, which are different from messages received by other organizations. This fact underlines the necessity of adjusting training sets to particular organization's needs.

4) Real-world environment

The current version of SpamKiller implemented at CERN to filter all incoming mails is using a dual-NBC structure described in this article. Previously it was based only on NBC_1 trained with *basic training set*, identical to this used in above tests. From the beginning of 2005, all the spams reported by CERN mail users automatically enlarge a set of spams used to train NBC_2.

So in fact every incoming mail is checked by two NBCs and a decision of any of them is in fact sufficient to treat an email as a spam. Figure 2 shows daily statistics of SpamKiller for February 2005.

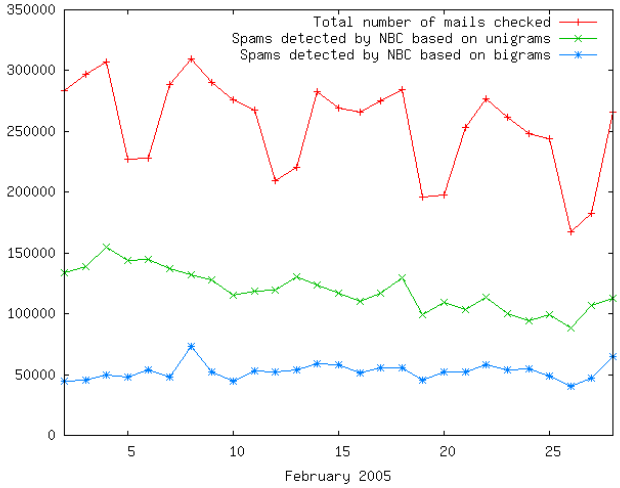

Fig. 2 Performance of naïve Bayesian classifiers working at CERN (February 2005)

It is worth noting that SpamKiller analyzes only part of the spam arriving at CERN SMTP gateways. The majority of it is rejected with early level mail classification based for example on *blacklists of internet addresses*. Only the less clear cases of spam are let in and subject to content analysis.

We can see that during the whole month the performance of NBC_1 is degrading. This is related to evolving nature of spam. We should not forget that this classifier is not being retrained and its knowledge is based on constant training set. Meanwhile the spam characteristic is changing, proving once again the need to dynamically train the classifier.

The latter classifier, based on bigrams, is constantly retrained with user-reported spams. The amount of detected spam is considerably lower than in the case of NBC_1. This is understandable as its conditions of classification are much stricter than in case of NBC_1. However, the amount of detected spam by NBC_2 is not decreasing; in fact is even slowly going up during the month. It indicates that the classifier is adapting to changing spam patterns and its knowledge is not aging.

Finally, it is worth mentioning that during more than two months of operations of NBC_2 there was not a single signal from the user side about possible *false-positives* whereas there were several cases of *false-positives* related to the work of NBC_1.And still the spam training set was at the beginning of March 2005 about two times larger than the training set containing legitimate emails. The training process can be therefore described as not needing operator assistance.

IV. CONCLUSION

The results show that retraining naïve Bayesian classifier based on single words can, apart from rising spam recall, deteriorate seriously a classification precision. In practice, this side effect should neglect such an option as a response to changing patterns of spam. However, supporting this classifier with another one, based on pairs of adjacent words can result in both recall increase and steady precision.

Of course, the additional classifier decisions are legitimate-biased. One of the reasons for this is a treatment of previously unseen bigrams, which are *a priori* considered as strongly legitimate-bound. This in turn allows us to safely break the balance between the strength of legitimate and spam training sets.

These features can allow creating an anti-spam system that will automatically incorporate knowledge about new spams reported by users. Running of such a system can be assured with minimal number of human interventions as a successful implementation at CERN Mail Services showed.

Nevertheless, still much could be done to improve the spam recall of the additional classifier. Areas of improvement should include parameterization (e.g. the length of feature vector) and fight with mail obfuscation which influences the decisions made by automatic classifiers in a negative way. In my opinion, whatever steps will be procured, the precision of the classifier should always be kept on as high level as possible.

To sum up, some questions concerning online retraining of

both of the classifiers will be discussed in the following sections.

1) Retraining NBC based on single words

Naïve Bayesian classifier can be easily retrained after initial training phase by adding new examples to either the group of legitimate mails or spams. However, there is little need to enlarge a set of legitimate mails as their category is in general static and is not subject to serious modifications with time. This is not the case of spam as this training set can be expanded with spam examples that are incorrectly classified by the classifier. It should be noted that retraining a NBC based on single words has serious drawbacks:

- A considerable amount of examples is needed to alter the probability of features (words) that were present in both training sets. This is contrary to the need of classifier adaptability as particular words present in newly observed spam will still be considered as legitimate or neutral during classification task.
- Increasing a spam training set too much may result in many neutral in reality features receiving a high spam probability just because of the fact that they were not present in the legitimate training set.
- Behind every word there is its semantics. It becomes evident that the most intelligent spam is constructed with as innocent as possible vocabulary which makes the informative content of a spam blurred and undefined.

Two first points are related to relatively small feature space. Of course, the second point one be addressed by equally adding examples to legitimate training set. Unfortunately, this can be difficult for reasons of privacy. Furthermore, simply extracting emails from user mailboxes may include a number of spams into the legitimate training set which in turn could heavily lower the classifier performance.

2) Retraining NBC based on bigrams

My experiments show that combining user feedback with naïve Bayesian classifier representing an email as set of bigrams (pairs of adjacent words) can largely eliminate problems mentioned in previous section. Such solution can be used as a complement of the NBC based on unigrams. Usefulness of pairs of words to represent an email is related to an extra dimension of feature space, which in turn largely increases the vocabulary size. Particular features (understood as pairs of adjacent words) have much lower probability of repeating itself when increasing the training sets. This fact makes it possible to easily extract 'fingerprints' of a new type of spam reported by users: usually it will contain a set of bigrams unseen in legitimate training set. Retraining the classifier with a certain number of examples of undetected spam will expand the dictionary with some word pairs with very high probability of representing a spam.

Moreover, because of adding a feature dimension, it is possible to drastically change the proportions of training sets. Amount of spam used to train the classifier can be now much higher than amount of solicited mails as particular 'spam' features will still be rarely met in solicited category.

Finally, such a classifier, from the theoretical point of view, will base its decisions more on a word structures used in an email than on a semantic meaning of particular words, which can be misleading. Spammers, as discussed before, tend to avoid using 'suspicious' words and special message structures with goal of making their spam as similar to an ordinary email as possible and to fool this way the anti-spam engines. Bigram representation could be regarded to a certain extent as an answer to spams constructed to be 'solicited-like' because it is more sensitive on particular phrases that can determine a spam in such cases.

V. From the Perspective of a Year

It is always an interesting experience to look back on an implementation and reevaluate its comportment and performance from the perspective of a longer period of time. As the original version of this article was submitted for conference presentation nearly a year ago, it is a tempting moment to review the SpamKiller spam statistics.

First of all it should be underlined that the CERN Mail Service is still using the same anti-spam framework in which the crucial role is reserved for naïve Bayesian classifiers. There were no major changes introduced to the algorithms and both of the versions of the classifiers, the one working on unigrams as well as the other based on bigrams, are in operation.

The classifier based on unigrams has not been retrained for over a year now, its training set remaining constant. Meanwhile, the version based on bigrams has continuously been retrained with spam examples reported by the CERN email users. This process has been proceeding automatically during the whole year and has been demanding only very little attention from *Internet Services* section staff members. Figure 3 presents some of the more recent statistics of the classifiers' performance. It is especially interesting to compare it with 14 months older data from Figure 2.

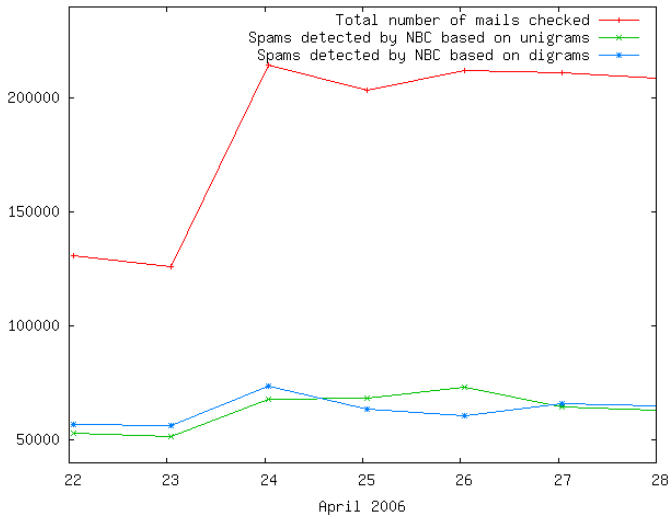

Fig. 3 Performance of naïve Bayesian classifiers working at CERN (22nd – 28th of April 2006)

On both figures we can observe the weekly fluctuations in total number of checked emails, which is related to lower users' activity during the weekends. However, major differences have occurred in the statistics of rejected spam. In the beginning of 2005, naïve Bayesian classifier based on unigrams was two to three times more effective that the one based on bigrams. Currently, their performance is very close in the number of detected spams and the decreasing tendency observed a year ago for NBC based on unigrams is not visible any more. Interestingly to mention, the performance of the classifier based on bigrams remains surprisingly stable at the level of about 50000 detected spams per day.

A question immediately arises whether both of the classifiers are redundant. The statistics suggest the negative answer. From the total number of spams detected by either of the classifiers about 40% is detected by both of them simultaneously. The remaining part is divided in approximately equal shares amongst them. This would suggest that both versions of naïve Bayesian classifiers are complementing each other, without a need of human supervision of the retraining process.

ACKNOWLEDGMENT

The author would like to warmly thank dr Paweł Cichosz for valuable comments on this paper. Special expressions of gratefulness must be directed towards *Internet Services* group at CERN and especially Mr. Emmanuel Ormancey (creator of SpamKiller) for making possible the completion of this analysis.

REFERENCES

[1] P.Graham. (2002, August). A Plan for Spam [Online]. Available: www.paulgraham.com/spam.html

[2] P. Graham, "Better Bayesian Filtering," in *Proceedings of Spam Conference 2003.* Available: http://spamconference.org/proceedings2003.html

[3] I.Androutsopoulos, J.Koutsias, K.V.Chandrinos, G.Paliouras, C.D. Spyropoulos, "An evaluation of naïve Bayesian anti-spam filtering," in *Workshop on Machine Training in the New Information Age 2000.*

[4] R.Segal, J.Crawford, J.Kephart, B.Leiba, "SpamGuru: An Enterprise Anti-Spam Filtering System," in *Proceedings of First Conference on Email and Anti-Spam (CEAS) 2004.*

[5] K. Aas, L. Eikvil. "Text categorization: A survey," Technical report, Norwegian Computing Center, 1999.

[6] G. Zipf, *Human Behavior and the Principle of Least Effort*. Addison-Wesley, 1949.

[7] C. M. Tan, Y. F. Wang, C. D. Lee, "The Use of BiGrams to Enhance Text Categorization," in *Journal Information Processing and Management.*, vol. 30, No. 4, pp. 529-546, 2002.

[8] H. Stern. "Optimizing Naïve Bayesian Networks for Spam Detection," *CSCI 6509: Natural Language Processing project*, Dalhousie University, Halifax, NS, Canada, 2002.

[9] T. Mitchell, *Machine learning*. McGraw Hill, 1997.

[10] N. Dalvi, P. Domingos, Mausam, S. Sanghai, D. Verma. "Adversarial Classification," in *Proceedings of the Tenth International Conference on Knowledge Discovery and Data Mining* (pp. 99-108), 2004.

[11] *Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies*. Request for Comments 2045, 1996.

[12] European Organization for Nuclear Research. Mail Service Web Site. http://mmm.cern.ch

[13] European Organization for Nuclear Research. Anti-Spam Web Site. http://mmmservices.web.cern.ch/mmmservices/Antispam/

[14] M. Fromberger. "Bayesian Classification of Unsolicited E-Mail," unpublished. Available: http://www.cs.dartmouth.edu/~sting/sw/bayes-spam.pdf

