Non-Parametric Spam Filtering based on kNN and LSA

Preslav Ivanov Nakov
Panayot Markov Dobrikov

Abstract. The paper proposes a non-parametric approach to filtering of unsolicited commercial e-mail messages, also known as spam. The email messages text is represented as an LSA vector, which is then fed into a kNN classifier. The method shows a high accuracy on a collection of recent personal email messages. Tests on the standard LINGSPAM collection achieve an accuracy of over 99.65%, which is an improvement on the best-published results to date.

Introduction

The amount of unsolicited commercial e-mails (also known as spam) has grown tremendously during the last few years and today it already represents the majority of the Internet traffic. Although the spam is normally easy to recognise and delete, doing so on an everyday basis is inconvenient. Once an e-mail address has entered in the widespread spam distribution lists it could become almost unusable unless some automated measures are taken.

Nowadays, it is largely recognised that the constantly changing spam form and contents requires filtering using machine learning (ML) techniques, allowing an automated training on up-to-date representative collections. The potential of learning has been first demonstrated by Sahami et al. [1] who used a Naïve Bayesian classifier. Several other researchers tried this thereafter and some specialised collections have been created to train ML algorithms. The most famous one LINGSPAM (a set of e-mail messages from the *Linguist List*) is accepted as a standard set for evaluating the potential of different approaches [2].

Text Categorization with *k*NN and LSA,

We used the *k*-nearest-neighbour classifier, which has been proved to be among the best performing text categorisation algorithms in many cases [3]. *k*NN calculates a similarity score between the document to be classified and each of the labelled documents in the training set. When *k* = 1 the class of the most similar document is selected. Otherwise, the classes of the *k* closest documents are used, taking into account their scores.

We combined *k*NN with *latent semantic analysis* (*LSA*). This is a popular technique for indexing, retrieval and analysis of textual data, and assumes a set of mutual latent dependencies between the terms and the contexts they are used in. This permits LSA to deal successfully with synonymy and partially with polysemy, which are the major problems with the word-based text processing techniques (due to the freedom and variability of expression). LSA is a two-stage process including learning and analysis. During the learning phase it is given a text collection and it produces a real-valued vector for each term and for each document. The second phase is the analysis when the proximity between a pair of documents or terms is calculated as the dot product between their normalised LSA vectors [4].

In our experiments, we built an LSA matrix (TF.IDF weighted) from the messages in the training set. The e-mail message to be classified is projected in the LSA space and then compared to each one from the training set. Then a *k*NN classifier for a particular value of *k* predicts its class.

Experiments and Evaluation

We used two document collections:

- *LINGSPAM* – 2,411 non-spam and 481 spam messages
- *personal collection* – 940 non-spam and 575 spam messages; it includes *all* email messages of one of the authors for a period of 5 consecutive weeks

While LINGSPAM is largely accepted as the ultimate test collection for spam filtering and allows for comparison between different approaches, it gives no practical idea of the *real* algorithm performance (just as any other frozen test collection). In a situation where the spam emails are constantly changing, so should do the test collections. A good solution is to use someone’s real emails for some period of time.

In this case it is very important to include *all* the emails no matter what they contain.

On LINGSPAM we measured the categorization accuracy using a stratified 10-fold cross validation. For the personal collection though, we adopted a more pragmatic approach:

- **Training.** We trained on *all* the email messages from the first 4 consecutive weeks: 766 non-spam and 453 spam.
- **Testing.** We tested on *all* emails from the subsequent fifth week: 174 non-spam and 122 spam.

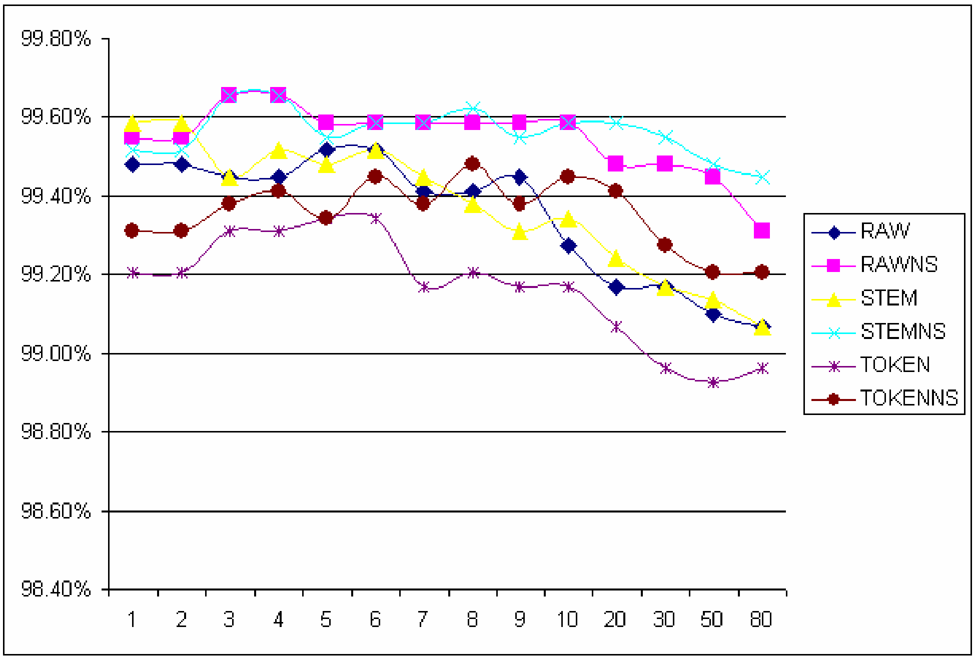

Figure 1. Accuracy on LINGSPAM as a function of the number of neighbours *k*, kNN.

We performed several experiments using as features: *words* as met in the text, *stems* as well as the *original tokens* as identified by the LINGSPAM author. Each of these features has been tried with and without stop-words removal. Although stripping the stop-words is beneficial for text categorisation in general, this is not always the case: e.g. they are very important features for authorship attribution (which is a text categorisation task). It was not clear to us whether they are good for spam filtering, but they were kept in the original LINGSPAM tokenisation, so we felt it was necessary to try either case. We ended up with the following features:

- RAW – raw words, as met in the text;
- RAWNS – like RAW but with stop-words filtered out;
- STEM – stemmed words (Porter stemmer);
- STEMNS – like STEM but with stop-words filtered out;
- TOKEN – original tokenisation as provided with LINGSPAM;
- TOKENNS – like TOKEN but with stop-words filtered out;

The results from a stratified 10-fold crossvalidation on LINGSPAM are shown on Figure 1, where the accuracy is drawn as a function of the number of neighbours *k* used in *k*NN. The highest accuracy of over 99.65% is achieved for moderate values of *k* such as 3 and 4.

The LINGSPAM tokenisation is richer than the other representations since it includes not only words but also some special symbols that can be good features for spam identification. Our experiments though, indicate this is not the case. In fact, as Figure 1 shows, TOKEN is the worst feature set. All the three features sets: TOKEN, RAW and STEM, perform much worse than their variants with stop-words removed (TOKENNS, RAWNS and STEMNS).

Discussion

Below we analyse the spam classifier's errors on the personal collection. Some newsletters have been classified as spam, although for other people they can be desired messages: e.g. *The Intel Developer Services News* and the *Bravenet News*. These examples show a well-known obstacle to the spam filtering techniques: a message that is absolutely valid for someone can be a spam for somebody else. However, in these two cases, there is no other newsletter from these two companies in the training set hence this is a true error. In general, once one or two issues of the newsletter are marked as spam by the user the following ones will be classified successfully.

A similar example is the following e-mail from the *Sun*'s *JCP* community, which has been annotated as spam in the testing set (while the receiver was subscribed), and as a non-spam by the classifier:

From: Stefan Hepper <sthepper@de.ibm.com>

To: JSR-168-EG@JCP.ORG
Subject: [SAP.JSR.168] Re: Not too late? GenericPortlet.render()
Hi Chris, the portal need to communicate the changed title to the portlet container, so that the portlet container can return a resource bundle that is correct for the current portlet instance. How would otherwise a portlet create a dynamic title …

Another misclassified e-mail was an obvious spam (an offer for enlargement of some organ). The reason why this happened was that we let the headers of the e-mail to be parsed. In this case, by coincidence there were many company e-mails from the address book in the CC list, which look similarity to other internal company e-mails.

Another error was a spam e-mail written in Russian (the only Cyrillic one). The classifier had no similar examples and should have made its decision based on the English header (not shown):

> Заберите у меня все, чем я обладаю, но оставьте мне мою речь, и скоро я обрету все, что имел Дэниэл Уэбстер Вся наша жизнь строится на общении — так устроено человеческое общество. Поэтому наибольших успехов в личной жизни...

Another ambiguous e-mail:

> Our friendship wont last that long?
> biyv gtdm zh vicd lns kjq alqnynuygbkojpzhpwkgbrmsvm

We now focus on the non-spam messages, which were classified as spam. This is a much more important case, as any such miss can invalidate the good results. There were three such e-mails:

- *Newsletter from ACM*

  Dear ACM TechNews Subscriber: Welcome to the September 15, 2003 edition of ACM TechNews, providing timely information for IT professionals three times a week. For instructions on how to unsubscribe from this service, please see below …

- *Google account confirmation*

  Welcome to Google Accounts. To activate your account and verify your e-mail address, please click on the following link: http://www.google.com/accounts/VE?c=8685708803317565504. If you have received this mail in error, please forward it to accounts-noreply@google.com …

- *Message after request for product download*

  Dear Mr. Dobrikov, Thank you for your recent download of WebLogic Platform 8.1 for Microsoft Windows (32 bit). Please click here to access installation and configuration guides to start you on your way using your download …

All the three cases exhibit some usual spam properties: phrases of the type “click here” or “download”, unsubscribe information etc., which mislead the classifier.

Conclusion and Future Work

We have shown that simple non-parametric ML methods such as kNN combined with LSA achieve state of the art accuracy on spam filtering. These results are encouraging and show the potential of the kNN+LSA combination. Note, that we used as features just raw words or stems from the message text and no other features, such as: subject, sender email, unsubscribe information, capitalisation, formatting etc., which are important knowledge sources and are used in most popular email filtering systems in operation. It was somewhat surprising to find that the exploitation of just the email text leads to an improvement on the best results on LINGSPAM. We believe the kNN classifier has a big potential and is a suitable candidate for spam filtering since, unlike the parametric approaches (e.g. Bayesian nets), it can learn from only few (often just one) examples. We plan to try kNN when using just non-word features as well as in combination with LSA.

References

[1] Sahami M., Dumais S., Heckerman D., Horvitz E. 1998. *A Bayesian Approach to Filtering Junk E-Mail*. In *Learning for Text Categorization: Papers from the 1998 Workshop. AAAI Technical Report WS-98-05.*

[2] Androutsopolos I., Paliouras G., Karkaletsis V., Sakkis G., Spyropoulos C., Stamatopoulos P. *Learning to filter spam e-mail: A comparison of a naive bayesian and a memory-based approach*. *Technical Report DEMO 2000/5, NCSR Demokritos, Greece*, 2000.

[3] Yang Y. *An evaluation of statistical approaches to text categorization*. *Journal of IR*, vol. 1, nos. 1/2, pp. 67-88, 1999.

[4] Landauer T., P. Foltz, D. Laham. *Introduction to LSA*. *Discourse Processes*, vol. 25, pp. 259-284, 1998.

Address:

Preslav Nakov, University of California, EECS, CS dept., Berkeley CA 94720, USA
nakov@cs.berkeley.edu

Panayot Dobrikov, SAP A.G, Neurottstrase 16, 69190 Walldorf, J2EE
panayot.dobrikov@sap.com