Learning to detect spam messages

Verónica Gil Costa, Marcelo Errecalde* and María Teresa Taranilla†
Universidad Nacional de San Luis.
San Luis, Argentina.
email{gvcosta, merreca, tarani}@unsl.edu.ar
VI Workshop de Agentes y Sistemas Inteligentes (WASI)

Abstract

The problem of unwanted e-mails (or *spam* messages) has been increasing for years. Different methods have been proposed in order to deal with this problem wich includes blacklists of known spammers, handcrafted rules and machine learning techniques.

In this paper we investigate the performance of the k Nearest Neighbours (k-NN) method in spam detection tasks. At this end, a number of different document codifications were tested. Moreover, we study how the vocabulary size reduction affects this task. In the experimental design, different k values were considered and results were analyzed with respect to a public mailing list and personal e-mail collections. The experiments showed that results with public mailing lists tend to be very optimistic and they should not be considered representative of those expected with personal user accounts.

Keywords: spam, anti-spam filtering, automated text categorization, machine learning, k-NN.

1 Introduction

The World Wide Web opened the Internet to many people by enabling access to information and services in a way that had never been possible before. As the Internet was expanded, the number of user was increased and consequently the "marketing" opportunities too.

The growing popularity and low cost of e-mail have attracted the attention of marketers. Using readily available bulk e-mail software and lists of e-mail addresses harvested from web pages and newsgroups archives, sending messages to millions of recipients is very easy and very cheap, and can be considered almost free. Consequently, these unsolicited e-mails bother users and fill their e-mails folders with unwanted messages.

Considering matters technically (but also with common sense) what is generally called "spam" is somewhat broader than the category "unsolicited commercial e-mail"; spam encompasses all the e-mail that we do not want and that is only very loosely directed at us. Such messages are not always commercial per se, and some push the limits of what it means to be solicited. Typically, when we refer to spam, we also imply bulk mails, because they are generally sent out in large batches, and also junk mails, because they are worthless to most recipients.

*Laboratorio de Investigación y Desarrollo en Inteligencia Computacional
†Proyecto Tecnologías Avanzadas de Bases de Datos, UNSL. Proyecto AL05_PF_0042 Geometría Computacional Universidad Politécnica de Madrid

As we think about the history of spam reduction we can see a gradual change in the approach over time, as the spam problem has changed. Many of us may think of spam as a new problem, but in fact, it goes back at least to 1975, as noted by Jon Postel [27]. At the start spam mostly referred to Usenet newsgroup posts that goes out of hand, wherein someone would post a message to hundreds of newsgroups, a message that was unrelated to most or all the newsgroups to which it was posted. Then, social and administrative action was sufficient: the perpetrator was castigated, perhaps privately, perhaps publicly; repeat offenders quickly be added to "kill lists". And so, early spam filtering simply identify "bad senders".

Spam-reduction techniques have developed rapidly over the last few years, as spam volumes has increased. We talk about spam reduction, because not always it is possible to eliminate the spam. This is partly because spammers, as they aggressively pursue their goals, always remain ahead of us in some areas. Still, with good techniques and customization we could come close to elimination.

From a technical point of view, spam filtering can be considered as a *text categorization* task, which is a well established field. Text categorization is the task of labelling natural language documents with thematic categories from a predefined set. In this context, spam filtering is a case of single-label categorization, i.e. the classification of incoming e-mails in two disjoint categories, the relevant (non-spam) and the irrelevant (spam).

In the last years, the dominant approach to automated text categorization is based on the application of *machine learning* techniques [24]. In this approach, a *classifier* is automatically derived from a inductive learning processes, which learns the correspondence between documents and categories, based on the evidence provided by a set of labelled documents (training set) [33]. Examples of this tendency includes Bayesian classifiers [17, 21], decision trees [18], nearest neighbours classification [36], neural networks [35], rule learning [2, 34], inductive learning algorithms [19, 7], maximum entropy models [25], boosting [31] and support vector machines [12, 13] among others.

The success of these techniques in text categorization has recently led researchers to explore the applicability of learning algorithms in anti-spam filtering [23, 26, 6, 28, 10].

One of the most used technique is the k Nearest Neighbours (k-NN) method [20]. Many researchers in text categorization have found that the k-NN algorithm achieves a very good performance in their experiments on different data sets [37, 4] and similar results have been obtained in spam filtering [28]. Given a set of labelled prototypes (i.e., text categories) and a test document to be classified, the k-NN method finds its k nearest neighbours among the training documents. The categories of the k neighbours are used to select the nearest category for the test document: each category gets the sum of votes of all the neighbours belonging to it and that one with the highest score is chosen. Other strategies calculate these scores taking into account the distances between the k neighbours and the test document or, alternatively, using a similarity measure like the scalar product. In this last strategy, which is the one used in our work, each document is represented through a vector of terms and each category gets a score equal to the sum of the similarities between the k neighbours and the test document.

This work investigates the performance of the k-NN method in spam detection tasks. At this end, a number of different document codifications were tested. Moreover, we study how the vocabulary size reduction affects this task. In the experimental design, different k values were considered and results were analyzed with respect to a public mailing list and personal e-mail collections. In this way, this paper extends previous works [23, 28] allowing to compare the performance of automatic learning with both kind of corpora. Besides, the results obtained and parameters used in this work are compared with others found in more general document categorization tasks in order to detect the peculiarities that arise when automatic learning is applied to spam detection.

The paper is organized as follows. Section 2 presents the alternative text codifications used in

the current work. Section 3 briefly describes the method utilized to perform the terms selection. In Section 4, the corpora used in this work are described. Section 5 shows the experimental design and the results obtained. Finally, in Section 6 the more relevant conclusions are presented.

2 Message Codification

In the present study, we used the conventional (real-valued) *vector space model* introduced by Salton [30] for the text codifications. The text of each message (e-mail) d was converted into a n-term vector $\vec{d} = \langle d_1, d_2, \ldots, d_n \rangle$, where n was the number of terms (words) which belong to the documents in the training set. The component d_i of vector $\vec{d}$ indicates how important the i-th term of vocabulary is in the document d.

The $TF \times IDF$ (*Term Frequency* $\times$ *Inverse Document Frequency*) weighting scheme was used for calculating the weight of terms (values of d_i) for a given document . $TF \times IDF$ gives a word higher weight if it is frequently appeared in a document and less frequently occurred across the document collection. The *term frequency* $TF_{d,i}$ of the i-th term of the document d is a text-specific statistic and it varies from one document to another, attempting to measure the importance of the term within a given document. On the other hand, the *Inverse Document Frequency* IDF_i is a global statistic and it characterises a given term i within an entire collection of N training documents. It is a measure of how widely the term i is distributed, and hence of how likely the term is to occur within any given document. The IDF metric is considered in order to punish those terms that occur in many of the documents of the collection and, therefore, are not relevant[1].

The weight of a term in a given document is usually *normalized* in a way that its importance depends on its frequency of occurrence with respect to the other terms of the same document, not on its absolute frequency of occurrence. Weighting a term by its absolute frequency would obviously tend to favour longer documents over shorter ones.

Below, the different alternatives for calculating and normalizing term weights are described. The SMART system conventional code scheme was used [29]. Each codification is composed by three letters: the first two letters refer, respectively, to the TF and IDF components, whereas the third one indicates whether normalization is employed or not[2].

SMART nomenclature

- d_i: It is the i-th component of vector $\vec{d} \in \mathbb{R}^n$.
- N: Number of training documents.
- $TF_{d,i}$: Term frequency (number of occurrences) of i-th term in the document d.
- DF_i: Document frequency of i-th term over the collection (number of documents where i is present).

Definition: $d_i = TF'_{d,i} IDF'_i NORM$

Where:

[1] If a term i occurs in the N documents of the collection, its IDF_i value is equal to 0.

[2] The cosine normalization is equivalent to converting the similarity function of the k-NN classifier into the calculation of the cosine between the two vectors, which is invariant with respect to the size of the two documents.

$TF'_{d,i}$ =

0 when $TF_{d,i} = 0$

If $TF_{d,i} \neq 0$ then

$n : none = TF_{d,i}$

$b : binary = 1$

$m : max - norm = \frac{TF_{d,i}}{max_i(TF_{d,i})}$

$a : aug - norm = 0.5 + 0.5\frac{TF_{d,i}}{max_i(TF_{d,i})}$

$l : log = 1 + \log(TF_{d,i})$

IDF'_i =

$n : none = 1$

$t : tfidf = \log(\frac{N}{DF_i})$

$NORM$ =

$n : none = 1$

$c : cosine = \frac{1}{\sqrt{\sum_i (TF'_{d,i} IDF'_{d,i})^2}}$

3 Dimensionality Reduction

In text categorization tasks the high dimensionality of the term space (i.e. the fact that the set of terms that occur at least once in the training set is large) may be problematic. The number of terms that occur in documents can be tens or hundred of thousands of terms for even a moderate-sized text collection. This is prohibitively high for many learning algorithms. Because of this, previous to classifier induction a pass of *dimensionality reduction* (DR) is often applied in order to reduce the dimensionality of the vector space. DR can also be beneficial since it tends to reduce *overfitting*[3].

The main purpose of a DR process is obtaining a list of terms that identify the collection, eliminating those terms with poor information. In some cases, DR can adopt very simple forms. As an example, a list of stop words is usually used to reduce the number of terms and it includes terms that do not provide any relevant information (typically, words as prepositions, articles, etc. [3]). Also, words occurring in less than a predefined number of messages are usually discarded.

Other more elaborated methods for selecting the terms to remove [38] include: Documents Frequency Thresholding [2], Information Gain [18], Mutual Information [32], Term Strength [39], etc. In our work, we employed the Information Gain (IG) method. IG measures the amount of information (number of bits) which contributes a term for the prediction of a category, as a function of its presence or absence in a given text. The IG value of a term i is defined to be:

$$\begin{aligned} IG_i = & -\sum_{j=1}^{m} \Pr(c_j) \log \Pr(c_j) \\ & + \Pr(i) \sum_{j=1}^{m} \Pr(c_j|i) \log \Pr(c_j|i) \\ & + \Pr(\neg i) \sum_{j=1}^{m} \Pr(c_j|\neg i) \log \Pr(c_j|\neg i) \end{aligned} \quad (1)$$

[3]This phenomenon is observed when a classifier is tuned also to the *contingent*, rather than just the *constitutive* characteristics of the training data.

where m is the number of existing categories, $\Pr(c_j)$ the probability that a text belongs to the category j, $\Pr(i)$ the probability of occurrence of the term i in the text, $\Pr(c_j|i)$ the probability that a text belongs to the category j given that the term i occurs in the text, and $\Pr(c_j|\neg i)$ is the probability that a text belongs to the category j given that the term i does not occur ($\neg i$ indicates no occurrence of the term i). Once calculated the IG_i value for all the terms, those terms with the highest values were selected.

4 Data Sets

As noted in [11] a common problem in spam-filtering research is the impossibility of direct comparison of experimental results from different researchers, as they are based on personal, different and not publicly available datasets [23, 26]. This problem is not present in other areas of text categorization where research has benefited significantly from the existence of publicly available, manually categorized document collections, like the Reuters-21578 collection [16], the 20 Newsgroups data set [15] and the WebKB data set [5], that have been used as standard benchmarks.

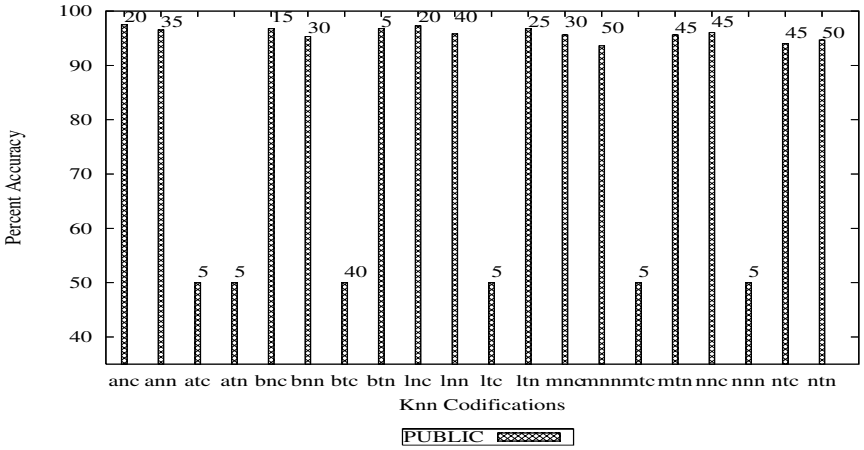
Producing similar corpora for anti-spam filtering is more complicated, because of privacy issues. Publicising spam messages does not pose a problem, since spam messages are distributed blindly to very large numbers of recipients, and, hence, they are effectively already publicly available. Legitimate messages, however, in general cannot be released without violating the privacy of their recipients and senders.

One way to bypass privacy problems is to experiment with legitimate messages collected from freely accessible newsgroups, or mailing lists with public archives. The Ling-Spam [1] corpus follows this approach. Ling-Spam is a mixture of spam messages, and legitimate messages sent via the Linguist list [4], a moderated and, hence, spam-free mailing list about the science and profession of linguistics. The corpus consists of 2893 messages:

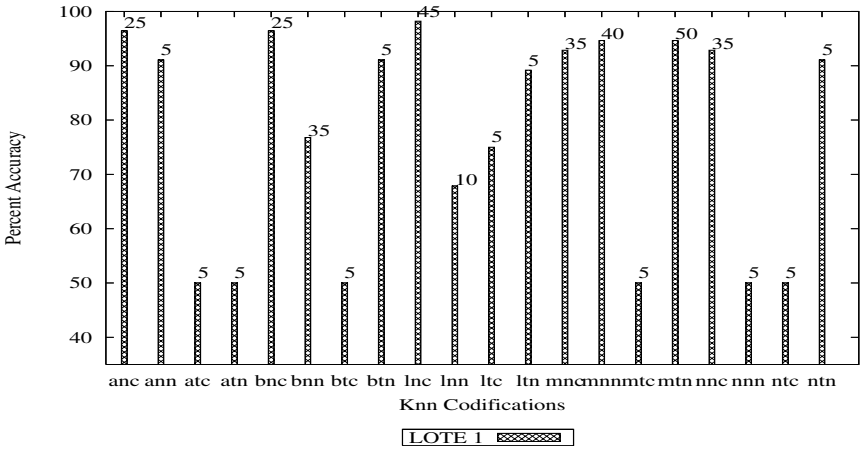
- 2412 legitimate messages, obtained by randomly downloading digests from the list's archives, breaking the digests into their messages, and removing text added by the list's server.
- 481 spam messages, received by Ion Androutsopoulos, one of the authors of the corpus. Attachments, HTML tags, and duplicate spam messages received on the same day have not been included.

The size of vocabulary of this corpus is 38517 words. Ling-Spam has the disadvantage that its legitimate messages are more topic-specific than the legitimate messages most users receive. Hence, the performance of a learning-based anti-spam filter on Ling-Spam may be an over-optimistic estimate of the performance that can be achieved on the incoming messages of a real user, where topic-specific terminology may be less dominant among legitimate messages. In that sense, Ling-Spam is more appropriate to experiments that explore the effectiveness of filters that guard against spam messages sent to topic-specific mailing lists [28].

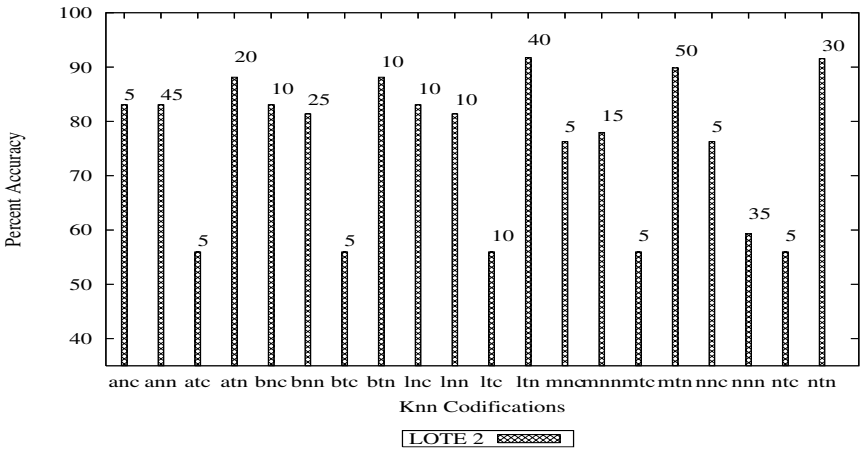
[4] The Linguist mailing list is archived at http://listserv.linguistlist.org/archives/linguist.html.



(a) Corpus *Public*



(b) Corpus *Lote 1*



(c) Corpus *Lote 2*

Figure 1: Best results of k-NN for different codifications

Although experimentation with public mailing lists is very important for comparative purposes, results obtained with personal mail folders should be considered. These user's real mail repositories are representative of the kind of data where learning methods could be used to adapt the spam filter to the personal preferences of each user. For this reason, our work includes experimentation with the Ling-Spam corpus and two collections named "LOTE 1" and "LOTE 2" obtained from personal e-mail folders of the authors. With this approach the performance of the k-NN method on personal data can be analyzed and also compared with results obtained with a public mailing list as Ling-Spam corpus. As far as we know, this kind of comparison have not being carried out in previous works about learning methods applied to spam filtering.

The "LOTE 1" data set contains 220 mails with a vocabulary size of 12600 words. The data are organized into two different groups, one corresponding to the spam mails and the other to the legitimate messages. The training set is composed by 160 messages and 60 messages are dedicated to be used as testing set.

The "LOTE 2" data set contains a total of 162 mails with a vocabulary size of 10321 words. They also are organized in two different groups (spam and no-spam) as in the previous case. A total of 102 messages are dedicated to the training set and 60 messages to the testing set.

None pre-processing operation was applied over these collections. The complete vocabulary includes all the words found in the mails under consideration.

5 Experimental Results

In this section, the experimental design and results obtained by the different experiments carried out with the three corpora explained above, are presented.
The mail categorization task was performed employing the k Nearest Neighbours method provided by the Rainbow system [22]. The results were averaged over 10 trials and different k values belonging to the set $\{5, 10, 15, 20, 25,30, 35, 40, 45, 50\}$ were considered. Furthermore, the reduction of vocabulary size was done by using the IG method, and the numbers of words eliminated of the vocabulary were 200, 500, 1000, 2000, 4000 and 8000.

Figure 1 shows the results obtained using the k-NN technique for different codifications with the three corpora presented above. The value over each bar denotes the lowest k found for the best result obtained with each codification. We can observe that values of k between 20 and 45 produce the best results. These values are similar to those reported in other text categorization tasks [14] where values of k close to 30 are recommended. The work presented in [28] reports smaller values of k, because the k neighborhood is taken to contain all the training instances at the *k closest distances*, rather than the k closest instances. As a result, if there is more than one neighbour at some of the k closest distances, the neighborhood contains more than k neighbours.

With the public corpus, the k-NN technique shows very different results depending on the codification considered. They are very good (the accuracy is over 90%) or they are really bad (with precision values under 55%).

Figure 2 shows the best values obtained for each codification with the three corpora. In all cases, the worst results (accuracy under 60%) corresponded to the *atc, btc, mtc* and nnn codifications and good results (accuracy over 80%) were observed when the *anc, ann, bnc, btn, lnc, ltn, mtn* and ntn codifications were used. From this, we can infer that similar results over different corpora are expected for a considerable number of codifications when the k-NN method is applied to spam detection tasks. This differs from previous works which address more general text categorization tasks and where the obtained results with each codification are very dependent on the particular data set used [9, 8].

With $*nc$ codifications the results are fairly good (the best are *anc*, *bnc* and *lnc*) and the behaviour of the classifier is good for $*tn$ codifications excepting for atn, where it only gets high values for the "LOTE 2" corpus. These results, confirm the importance of the IDF information ($*t*$ codifications) which sub-estimates those terms that occur in many texts and are not relevant, and of the cosine normalization ($**c$ codifications) which weights a term in a given text with respect to the other terms of the same text and not on its absolute frequency of occurrence. However, the combination of both characteristics is not a warranty of good results as can be observed with $*tc$ codifications which generally produced bad results, except for the ntc codification with the public corpus.

Other works [8, 9] in general text categorization usually have obtained good results for the codifications $*tc$ when standard corpora like the 20 Newsgroup have been used. However, this performance was not achieved with corpora that are not as richer from a syntactic and semantic point of view as the 20 Newsgroup corpus [5], or when some kind of noise was introduced in the categorization of the documents in the training set [9]. In these cases the best codifications were anc, lnc y bnc with the WebKB corpus, meanwhile the codifications $*tn$ (ltn, mtn, atn) and ann produced the best results when noise was introduced in the training examples.

Figures 3, 4 and 5 show the impact of vocabulary size reduction for the *anc*, *bnc*, *lnc* and btn codifications. The k-NN method with the public corpus has an uniform behavior and the vocabulary size reduction has not a great impact on the performance. As expected, when the number of words to remove is increased, the accuracy of filter decreases (with less variety of words, less probability of detecting spam messages). This kind of results are common in general text categorization tasks with standard corpora as the 20 Newsgroup [8, 9] where usually is observed a continuous decrease of the error percentage when the vocabulary size increases.

This behaviour is not observed with the "LOTE 1" and "LOTE 2" corpora because the goodness of classifier do not always decrease when the vocabulary size is reduced. With bnc codification for example, the accuracy percentage increases from a reduction of 200 words to 500 words. This anomaly has also been observed with poorly structured corpora and when noise is introduced in the categorization of documents of training set [9, 8].

Finally, Figure 6 compares the performance of Support Vector Machine (svm), Naive Bayes and the k-NN techniques. As can be observed, the svm method obtain the worst results over the three corpora, with respect to the other techniques. The Naives Bayes results are comparable with those obtained with k-nn for the public corpus, but k-NN outperforms Naive Bayes over privates corpora. According to these results, described in more detail in Table 1, we can conclude that k-NN exhibits a better performance than the other techniques over the three corpora considered.

6 Conclusions

In this paper, we investigated the performance of the k-NN technique in spam filtering tasks. At this end, different document codifications, k values and vocabulary sizes were considered. In the experimental design, the results were analyzed with respect to a public mailing list and personal e-mail collections obtained from the authors.

The experiments showed that in this domain goods results are obtained with values of k similar to those used in more general text categorization tasks.

The behaviour of k-NN method when it was applied over the public mailing list was more accurate and predictable than when personal mail accounts were used. This is due to the fact that the preferences varies depending on the user and it make more difficult the automatic spam detection.

[5] This is the case of the WebKB corpus composed by web pages.

	Naives Bayes	K-NN	svm
Lote 1	96.43	98.21 (lnc)	94.64
Lote 2	91.53	91.73 (ltn)	91.41
Public	96.96	97.28 (lnc)	91.53

Table 1: Percent Accurancy for each technique.

In this sense, the performance of k-NN over the Ling-Spam corpus was very similar to its performance over standard corpora observed in previous works, in experiments with different codifications and reduction of vocabulary size. In contrast, results on personal mail collections seem to be closer to those observed when less structured corpora are considered. Consequently, the experiments with public mailing lists tend to be very optimistic and they should not be considered representative of the results that can be expected with personal user accounts.

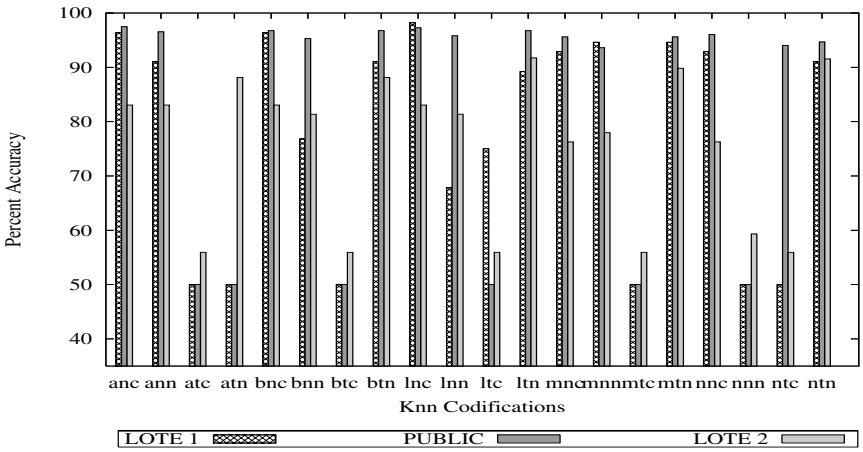
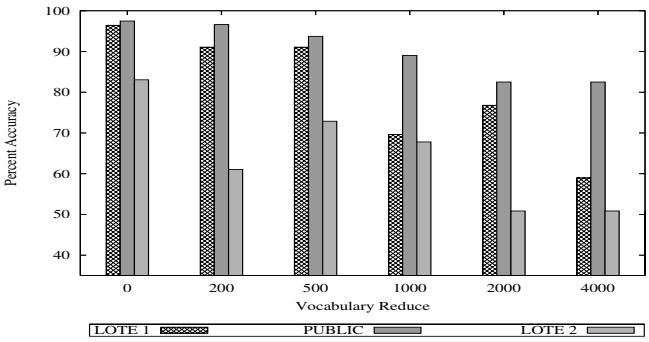
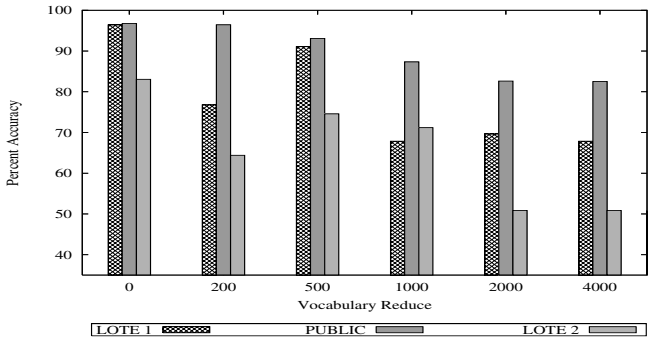


Figure 2: Results on three corpora (best k value selected)



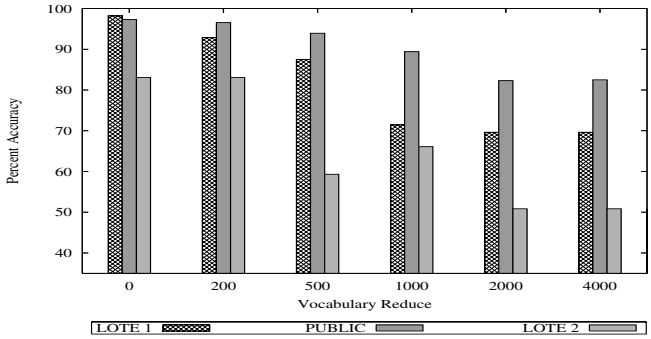
(a) *anc* codification

Figure 3: Vocabulary reduction impact for the anc codification

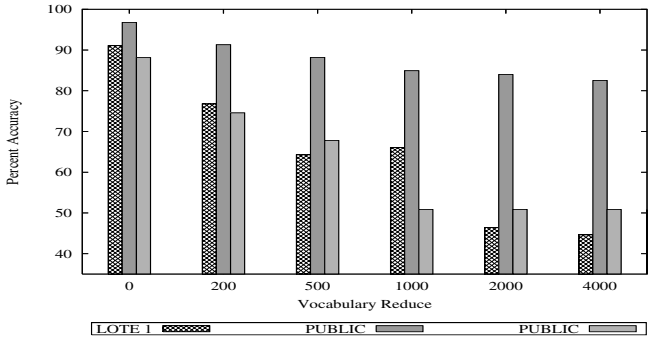


(a) *bnc* codification

Figure 4: Vocabulary reduction impact for the bnc codification



(a) *lnc* codification



(b) *btn* codification

Figure 5: Vocabulary reduction impact for the selected codifications

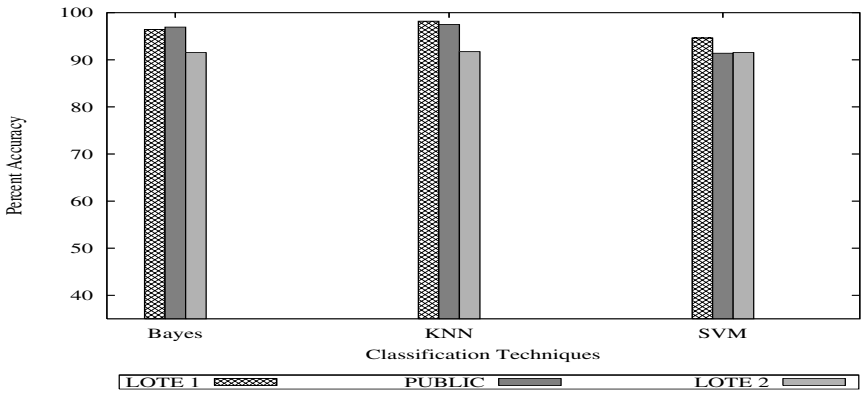


Figure 6: Bayes-SVM-KNN

References

[1] I. Androutsopoulos. Ling-spam corpus. Available at http://www.iit.demokritos.gr/skel/i-config/., 2000.

[2] C. Apté, F. Damerau, and S. M. Weiss. Automated learning of decision rules for text categorization. *ACM Trans. Inf. Syst.*, 12(3):233–251, 1994.

[3] R. Baeza-Yates and B. Ribeiro-Neto. *Modern Information Retrieval*. Addison Wesley, England, 1999.

[4] L. Baoli, C. Yuzhong, and Y. Shiwen. A comparative study on automatic categorization methods for chinese search engine. In *Proceedings of 8th Joint International Computer Conference*, pages 117–120, Hangzhou, Zhejiang, 2002.

[5] M. Craven, D. DiPasquo, D. Freitag, A. McCallum, T. Mitchell, K. Nigam, and S. Slattery. Learning to extrac symbolic knowledge from the world wide web. In *15th Conference of the American Association for Artificial Intelligence*, pages 509–516, 1998.

[6] H. Drucker, D. Wu, and V. Vapnik. Support vector machines for spam categorization. *IEEE Transactions on Neural Networks*, 10(5):1048–1054, 1999.

[7] S. Dumais, J. Platt, D. Heckerman, and M. Sahami. Inductive learning algorithms and representations for text categorization. In *CIKM '98: Proceedings of the 7th international conference on Information and knowledge management*, pages 148–155, New York, NY, USA, 1998. ACM Press.

[8] E. Ferretti, M. Errecalde, and P. Rosso. The influence of semantics in text categorisation: A comparative study using the k nearest neighbours method. In *Accepted to be published in the Proceedings of the 2nd Indian International Conference on Artificial Intelligence (IICAI-05)*, 2005.

[9] E. Ferretti, J. Lafuente, and P. Rosso. Semantic text categorization using the k nearest neighbours method. In *1st Indian International Conference on Artificial Intelligence*, pages 434–442, 2003.

[10] J. G. Hidalgo and M. M. Lopez. Combining text and heuristics for cost-sensitive spam filtering. In *Proceedings of the 4th Computational Natural Language Learning Workshop.*, pages 99–102, Lisbon, Portugal, 2000.

[11] G. P. I. Androutsopoulos and E. Michelakis. Learning to filter unsolicited commercial e-mail. TR 2004/2. Technical report, Greek National Centre for Scientific Research "Demokritos", 2004. revised version.

[12] T. Joachims. Text categorization with suport vector machines: Learning with many relevant features. In *ECML '98: Proceedings of the 10th European Conference on Machine Learning*, pages 137–142, London, UK, 1998.

[13] T. Joachims. Transductive inference for text classification using support vector machines. In *ICML '99: Proceedings of the 16th International Conference on Machine Learning*, pages 200–209, San Francisco, CA, USA, 1999.

[14] LaFuente and Alfons. Comparación de codificaciones de documentos para la clasificación con k vecinos mas próximos. *JOTRI Conference*, pages 37–42, 2005.

[15] K. Lang. 20 newsgroups, the original data set. http://kdd.ics.uci.edu/databases/20newsgroups/20newsgroups.html.

[16] D. D. Lewis. Reuters-21578. A text categorization test collection, http://www.daviddlewis.com/resources/testcollections/reuters21578, 1996.

[17] D. D. Lewis. Naive (Bayes) at forty: The independence assumption in information retrieval. In *ECML '98: Proceedings of the 10th European Conference on Machine Learning*, pages 4–15, London, UK, 1998. Springer-Verlag.

[18] D. D. Lewis and M. Ringuette. A comparison of two learning algorithms for text classification. In *Proceedings of the 3rd Annual Symposium on Document Analysis and Information Retrieval*, pages 81–93, 1994.

[19] D. D. Lewis, R. E. Schapire, J. P. Callan, and R. Papka. Training algorithms for linear text classifiers. In *SIGIR '96: Proceedings of the 19th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 298–306, New York, NY, USA, 1996. ACM Press.

[20] C. D. Manning and H. Schtüze. *Foundations of Statistical Natural Language Processing*. The MIT Press, 1999.

[21] A. McCallum and K. Nigam. A comparison of event models for naive bayes text classification. In *Proceedings of AAAI-98 Workshop on Learning for Text Categorization*, 1998.

[22] A. K. McCallum. Bow: A toolkit for statistical language modeling, text retrieval, classification and clustering. http://www.cs.cmu.edu/∼mccallum/bow, 1996.

[23] D. H. Mehran Sahami, Susan Dumais and E. Horvitz. A bayesian approach to filtering junk e-mail. In *Proceeding of AAAI-98 Workshop on Learning for Text Categorization*, pages 55–62, Madison Wisconsin, 1998.

[24] T. M. Mitchell. *Machine Learning*. McGraw-Hill, 1997.

[25] K. Nigam, J. Lafferty, and A. McCallum. Using maximum entropy for text classification. In *Proceedings of IJCAI-99 Workshop on Machine Learning for Information Filtering*, pages 61–67, 1999.

[26] P. Pantel and D. Lin. Spamcop– a spam classification and organization program. In *Proceeding of AAAI-98 Workshop on Learning for Text Categorization*, pages 95–98, Madison Wisconsin, 1998.

[27] J. Postel. On the junk mail problem. *Network Working Group Request for Comments*, Nov. 1975.

[28] G. Sakkis and I. A. et. al. A memory-based approach to anti-spam filtering for mailing lists. *Information Retrieval*, 6(1):49–73, 2003.

[29] G. Salton. *The Smart Retrieval System: Experiments in Automatic Document Processing*. Prentice Hall, 1971.

[30] G. Salton and C. Buckley. Term-weighting approaches in automatic text retrieval. *Information Processing and Management*, 24(5):513–523, 1988.

[31] R. E. Schapire and Y. Singer. BoosTexter: A boosting-based system for text categorization. *Machine Learning*, 39(2/3):135–168, 2000.

[32] H. Schütze, D. A. Hull, and J. O. Pedersen. A comparison of classifiers and document representations for the routing problem. In *SIGIR '95: Proceedings of the 18th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 229–237, New York, NY, USA, 1995. ACM Press.

[33] F. Sebastiani. Machine learning in automated text categorization. Revised Version of Technical Report IEI–B4–31–1999, Consiglio Nazionale delle Ricerche, Italy, 2001.

[34] S. Slattery and M. Craven. Combining statistical and relational methods for learning in hypertext domains. In *ILP '98: Proceedings of the 8th International Workshop on Inductive Logic Programming*, pages 38–52, London, UK, 1998. Springer-Verlag.

[35] E. D. Wiener, J. O. Pedersen, and A. S. Weigend. A neural network approach to topic spotting. In *Proceedings of SDAIR-95, 4th Annual Sym-posium on Document Analysis and Information Retrieval*, pages 317–332, 1995.

[36] Y. Yang. Expert network: effective and efficient learning from human decisions in text categorization and retrieval. In *SIGIR '94: Proceedings of the 17th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 13–22, New York, NY, USA, 1994. Springer-Verlag New York, Inc.

[37] Y. Yang and X. Liu. A re-examination of text categorization methods. In *22nd Annual International SIGIR*, pages 42–49, 1999.

[38] Y. Yang and J. O. Pedersen. A comparative study on feature selection in text categorization. In *ICML '97: Proceedings of the 14th International Conference on Machine Learning*, pages 412–420, San Francisco, CA, USA, 1997. Morgan Kaufmann Publishers Inc.

[39] Y. Yang and J. Wilbur. Using corpus statistics to remove redundant words in text categorization. *J. Am. Soc. Inf. Sci.*, 47(5):357–369, 1996.