In: *Proceedings of CoNLL-2000 and LLL-2000*, pages 99–102, Lisbon, Portugal, 2000.

Combining Text and Heuristics for Cost-Sensitive Spam Filtering

José M. Gómez Hidalgo
Universidad Europea–CEES, Spain
jmgomez@dinar.esi.uem.es

Manuel Maña López*
Universidad de Vigo, Spain
mjlopez@uvigo.es

Enrique Puertas Sanz
Universidad Europea–CEES, Spain
epsilonmail@retemail.es

Abstract

Spam filtering is a text categorization task that shows especial features that make it interesting and difficult. First, the task has been performed traditionally using heuristics from the domain. Second, a cost model is required to avoid misclassification of legitimate messages. We present a comparative evaluation of several machine learning algorithms applied to spam filtering, considering the text of the messages and a set of heuristics for the task. Cost-oriented biasing and evaluation is performed.

1 Introduction

Spam, or more properly Unsolicited Commercial E-mail (UCE), is an increasing threat to the viability of Internet E-mail and a danger to Internet commerce. UCE senders take away resources from users and service suppliers without compensation and without authorization. A variety of counter-measures to UCE have been proposed, from technical to regulatory (Cranor and LaMacchia, 1998). Among the technical ones, the use of filtering methods is popular and effective.

UCE filtering is a text categorization task. Text categorization (TC) is the classification of documents with respect to a set of one or more pre-existing categories. In the case of UCE, the task is to classify e-mail messages or newsgroups articles as UCE or not (that is, legitimate). The general model of TC makes use of a set of preclassified documents to classify new ones, according to the text content (i.e. words) of the documents (Sebastiani, 1999).

Although UCE filtering seems to be a simple instance of the more general TC task, it shows two special characteristics:

- First, UCE filtering has been developed using very simple heuristics for many years. For example, one individual could manually build a filter that classifies as "spam" messages containing the phrase "win big money", or with an unusual (big) number of capital letters or non-alphanumeric characters. These rules are examples of simple but powerful heuristics that could be used to complement a word-based automatic TC system for UCE filtering.
- Second, all UCE filtering errors are not of equal importance. Individuals usually prefer conservative filters that tend to classify UCE as legitimate, because missing a legitimate message is more harmful than the opposite. A cost model is imperative to avoid the risk of missing legitimate e-mail.

Many learning algorithms have been applied to the problem of TC (Yang, 1999), but much less with the problem of UCE filtering in mind. Sahami and others (1998) propose the utilization of a Naive Bayes classifier based on the words and a set of manually derived heuristics for UCE filtering, showing that the heuristics improve the effectiveness of the classifier. Druker and others (1999) compare boosting, Support Vector Machines, Ripper and Rocchio classifiers for UCE filtering. Andruotsopoulos and others (2000) present a cost-oriented evaluation of the Naive Bayes and k-nearest neighbor (kNN) algorithms for UCE filtering. Finally, Provost (1999) compares Naive Bayes and RIPPER for the task. These three last works do not consider any set of heuristics for UCE filtering. So, an extensive evaluation of learning algorithms combining words and heuristics

* Partially supported by the CICYT, project no. TEL99-0335-C04-03

remains to be done. Also, although the evaluations performed in these works have taken into account the importance of misclassifying legitimate e-mail, they have not considered that many learning algorithms (specially those that are error-driven) can be biased to prefer some kind of errors to others.

In this paper, we present a comparative evaluation of a representative selection of Machine Learning algorithms for UCE filtering. The algorithms take advantage of two kinds of information: the words in the messages and a set of heuristics. Also, the algorithms are biased by a cost weighting schema to avoid, if possible, misclassifying legitimate e-mail. Finally, algorithms are evaluated according to cost-sensitive measures.

2 Heuristics for UCE classification

Sahami and others (Sahami et al., 1998) have proposed a set of heuristic features to complement the word Bayesian model in their work, including: a set of around 35 hand-crafted key phrases (like "free money"); some non text features (like the domain of the sender, or whether the message comes from a distribution list or not); and features concerning the non-alphanumeric characters in the messages.

For this work, we have focused in this last set of features. The test collection used in our experiments, Spambase, already contained a set of nine heuristic features. Spambase[1] is an e-mail messages collection containing 4601 messages, being 1813 (39%) marked as UCE. The collection comes in preprocessed (not raw) form, and its instances have been represented as 58-dimensional vectors. The first 48 features are words extracted from the original messages, without stop list nor stemming, and selected as the most unbalanced words for the UCE class. The next 6 features are the percentage of occurrences of the special characters ";", "(", "[", "!", "$" and "#". The following 3 features represent different measures of occurrences of capital letters in the text of the messages. Finally, the last feature is the class label. So, features 49 to 57 represent heuristic attributes of the messages.

In our experiments, we have tested several learning algorithms on three feature sets: only words, only heuristic attributes, and both. We have divided the Spambase collection in two parts: a 90% of the instances are used for training, and a 10% of the messages are retained for testing. This split has been performed preserving the percentages of legitimate and UCE messages in the whole collection.

3 Cost-sensitive UCE classification

According to the problem of UCE filtering, a cost-sensitive classification is required. Each learning algorithm can be biased to prefer some kind of missclassification errors to others. A popular technique for doing this is resampling the training collection by multiplying the number of instances of the preferred class by the cost ratio. Also, the unpreferred class can be downsampled by eliminating some instances. The software package we use for our experiments applies both methods depending on the algorithm tested.

We have tested four learning algorithms: Naive Bayes (NB), C4.5, PART and k-nearest neighbor (kNN), all implemented in the Weka package (Witten and Frank, 1999). The version of Weka used in this work is Weka 3.0.1. The algorithms used can be biased to prefer the mistake of classify a UCE message as not UCE to the opposite, assigning a penalty to the second kind of errors. Following (Androutsopoulos et al., 2000), we have assigned 9 and 999 (9 and 999 times more important) penalties to the missclassification of legitimate messages as UCE. This means that every instance of a legitimate message has been replaced by 9 and 999 instances of the same message respectively for NB, C4.5 and PART. However, for kNN the data have been downsampled.

4 Evaluation and results

The experiments results are summarized in the Table 1, 2 and 3. The learning algorithms Naive Bayes (NB), 5-Nearest Neighbor (5NN), C4.5 and PART were tested on words (-W), heuristic features (-H), and both (-WH). The kNN algorithm was tested with values of k equal to 1, 2, 5 and 8, being 5 the optimal number of neighbors. We present the weighted accuracy (*wacc*), and also the recall (*rec*) and precision (*pre*) for the class UCE. Weighted accuracy is a measure that weights higher the hits and misses

[1] This collection can be obtained from http://www.ics.uci.edu/mlearn/MLRepository.html.

for the preferred class. Recall and precision for the UCE class show how effective the filter is blocking UCE, and what is its effectiveness letting legitimate messages pass the filter, respectively (Androutsopoulos et al., 2000).

In Table 1, no costs were used. Tables 2 and 3 show the results of our experiments for cost ratios of 9 and 999. For these last cases, there were not enough training instances for the kNN algorithm to perform classification, due to the downsampling method applied by Weka.

5 Discussion and conclusions

The results of our experiments show that the best performing algorithms are C4.5 and Part. However, for the cost value of 999, both algorithms degrade to the trivial rejector: they prefer to classify every message as legitimate in order to avoid highly penalized errors. With these results, neither of these algorithms seems useful for autonomous classification of UCE as stated by Androutsopoulos, since this cost ratio represents a scenario in which UCE messages are deleted without notifying the user of the UCE filter. Nevertheless, Part-WH shows competitive performance for a cost ratio of 9. Its numbers are comparable to those shown in a commercial study by the top performing Brightmail filtering system (Mariano, 2000), which reaches a UCE recall of 0.73, and a precision close to 1.0, and it is manually updated.

Naive Bayes has not shown high variability with respect to costs. This is probably due to the sampling method, which only slightly affects to the estimation of probabilities (done by approximation to a normal distribution). In (Sahami et al., 1998; Androutsopoulos et al., 2000), the method followed is the variation of the probability threshold, which leads to a high variation of results. In future experiments, we plan to apply the uniform method MetaCost (Domingos, 1999) to the algorithms tested in this work, for getting more comparable results.

With respect to the use of heuristics, we can see that this information alone is not competitive, but it can improve classification based on words. The improvement shown in our experiments is modest, due to the heuristics used. We are not able to add other heuristics in this case because the Spambase collection comes in a preprocessed fashion. For future experiments, we will use the collection from (Androutsopoulos et al., 2000), which is in raw form. This fact will enable us to search for more powerful heuristics.

References

I. Androutsopoulos, G. Paliouras, V. Karkaletsis, G. Sakkis, C.D. Spyropoulos, and P. Stamatopoulos. 2000. Learning to filter spam e-mail: A comparison of a naive bayesian and a memory-based approach. Technical Report DEMO 2000/5, Inst. of Informatics and Telecommunications, NCSR Demokritos, Athens, Greece.

Lorrie F. Cranor and Brian A. LaMacchia. 1998. Spam! *Comm. of the ACM*, 41(8).

Pedro Domingos. 1999. Metacost: A general method for making classifiers cost-sensitive. In *Proc. of the 5th International Conference on Knowledge Discovery and Data Mining.*

Harris Drucker, Donghui Wu, and Vladimir N. Vapnik. 1999. Support vector machines for spam categorization. *IEEE Trans. on Neural Networks*, 10(5).

Gwendolin Mariano. 2000. Study finds filters catch only a fraction of spam. *CNET News.com.* Available at http://news.cnet.com/news/0-1005-200-2086887.html.

Jefferson Provost. 1999. Naive-bayes vs. rule-learning in classification of email. Technical Report available at http://www.cs.utexas.edu/users/jp/research/, Dept. of Computer Sciences at the U. of Texas at Austin.

Mehran Sahami, Susan Dumais, David Heckerman, and Eric Horvitz. 1998. A bayesian approach to filtering junk e-mail. In *Learning for Text Categorization: Papers from the 1998 Workshop*. AAAI Tech. Rep. WS-98-05.

Fabrizio Sebastiani. 1999. A tutorial on automated text categorisation. In *Proc. of the First Argentinian Symposium on Artificial Intelligence (ASAI-99).*

Ian H. Witten and Eibe Frank. 1999. *Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations.* Morgan Kaufmann.

Yiming Yang. 1999. An evaluation of statistical approaches to text categorization. *Information Retrieval*, 1(1-2).

classifier	*rec*	*pre*	*wacc*	*classifier*	*rec*	*pre*	*wacc*
NB-W	0.97	0.74	0.85	C4.5-W	0.78	0.87	0.86
NB-H	0.31	0.80	0.69	C4.5-H	0.81	0.90	0.88
NB-WH	0.97	0.73	0.84	C4.5-WH	0.85	0.89	0.89
5NN-W	0.79	0.85	0.86	Part-W	0.81	0.87	0.87
5NN-H	0.72	0.83	0.83	Part-H	0.73	0.86	0.84
5NN-WH	0.75	0.87	0.85	Part-WH	0.89	0.91	0.92

Table 1: UCE recall, UCE precision and weighted accuracy for costs = 1.

classifier	*rec*	*pre*	*wacc*	*classifier*	*rec*	*pre*	*wacc*
NB-W	0.97	0.74	0.78	C4.5-W	0.55	0.96	0.95
NB-H	0.23	0.76	0.90	C4.5-H	0.41	0.96	0.95
NB-WH	0.97	0.74	0.78	C4.5-WH	0.71	0.96	0.96
5NN-W	–	–	–	Part-W	0.59	0.98	0.96
5NN-H	–	–	–	Part-H	0.23	0.93	0.93
5NN-WH	–	–	–	Part-WH	0.71	0.98	0.97

Table 2: UCE recall, UCE precision and weighted accuracy for costs = 9.

classifier	*rec*	*pre*	*wacc*	*classifier*	*rec*	*pre*	*wacc*
NB-W	0.18	0.79	0.96	C4.5-W	0.00	0.00	0.99
NB-H	0.23	0.76	0.90	C4.5-H	0.00	0.00	0.99
NB-WH	0.97	0.74	0.78	C4.5-WH	0.00	0.00	0.99
5NN-W	–	–	–	Part-W	0.00	0.00	0.99
5NN-H	–	–	–	Part-H	0.00	0.00	0.99
5NN-WH	–	–	–	Part-WH	0.00	0.00	0.99

Table 3: UCE recall, UCE precision and weighted accuracy for costs = 999.