www.ijbar.org
ISSN 2249-3352 (P) 2278-0505 (E)
Cosmos Impact Factor-5.86

Methods for Identifying Spam using Text Clustering

[1] M. Nagshwarappa, [2] G. Shireesha,

[1]Assistant Professor, Megha Institute of Engineering & Technology for Women, Ghatkesar.
[2] MCA Student, Megha Institute of Engineering & Technology for Women, Ghatkesar.

Abstract

Using a vector space model for text clustering, we provide a novel spam detection method. In order to automatically extract the cluster description, our technique uses a spherical k-means algorithm to automatically create disjoint clusters for all spam and non-spam emails. We then acquire the centroid vectors of these clusters. To determine whether an email is considered spam or not, we count all of the spam emails in the cluster and apply a label to each centroid vector accordingly. The centroid vector and the new mail vector are compared using the cosine similarity calculation when fresh mail arrives. The last step is to give the new email the label of the cluster that is most applicable. In order to effectively identify spam emails, our approach can extract a wide variety of subjects from both spam and non-spam messages. We provide our spam detection method and our experimental results using the Ling-Spamtest collection in this publication.

1. Introduction

Internet users have been increasingly frustrated in recent years by spam, which is officially known as Unsolicited Bulk Email (UBE). Spam email is sent to a huge number of people indiscriminately because it is inexpensive to send. As a result of the time and effort required to distinguish between spam and legitimate email, the mail server may experience overload when dealing with a high volume of spam messages. Several client-side spam detection and filtering initiatives have been launched in an effort to combat the spam issue. Bayesian classifiers such as Naive Bayes[1,3,7,11], C4.5[10], Ripper[4], and Support Vector Machine (SVM)[6,9] are among the numerous Machine Learning (ML) techniques that have been used to this issue in earlier studies. Bayesian classifiers were successful in these methods, leading to their widespread use in a variety of filtering applications. Nonetheless, almost all methods study spam and non-spam communications separately to determine the feature set's distribution. There are many different kinds of spam email that circulate nowadays. Some examples include ads that aim to make money or sell something, urban legends that propagate false information, etc. In addition, some HTML emails include web bugs, which are images included in email messages with the purpose of tracking the recipient's activity. Thus, even with the current filtering methods, some spam emails are still classified as legitimate. This study presents a novel method for spam identification based on text clustering and the vector space model. This method is more effective in building the spam detection model based on the contents of different kinds of emails. the ly. For all emails, spam and otherwise, the system automatically uses a spherical-means algorithm[5] to calculate disjoint clusters, and then it gets the centroid vectors of those clusters so it can extract their descriptions. We determine the label ('spam' or 'non-spam') for each centroid vector by computing the amount of spam emails in the cluster. The cosine similarity between the fresh mail vector and the centroid vector is computed when new mail arrives. The last step is to give the new email the label of the cluster that is most applicable. Our approach can efficiently identify spam emails and extract various subjects from both spam and non-spam emails. We provide our spam detection method and the results of our trials using the Ling-Spam[1, 2, 12] test collection in this work.

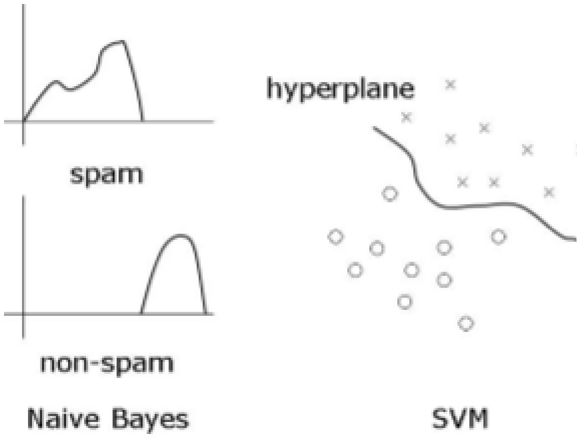


Figure 1. spam detection model using Naive Bayes and SVM

2. Motivation

No system can reliably identify all spam since spammers send out different forms of spam. Ads to pornographic websites have been flooding our inbox recently. Drugads, chainletters, and urban legends are just a few forms of spam that have emerged recently. A number of filtering programs make use of Bayesian classifiers[3] to learn and discover the two distributions of spam and non-spam. Thus, we get satisfactory results for standard spam material, but they are ineffective for the vast majority of spam, which appears just once every several months. Consequently, as the diversity of spam messages grows, it becomes progressively difficult to identify all of them using a single distribution. It is very desired to dynamically update the model for freshly received mail in addition to building a static model from all the training data for the system to recognize different types of spam. It is essential to train the system to accurately evaluate freshly received mail if it incorrectly evaluates it using the current model. But learning all the emails with the new messages and building the new model took a long time. The model may be rebuilt in a very short amount of time if the total number of emails is quite modest. In most cases, however, the volume of emails makes rapid model construction impossible. In addition, the prior model formulation did not necessarily include all spam for the user. Hence, it's very probable that the preceding model cannot be updated correctly. To boost performance, the model should include readily updatable features like incremental text classification and relevance feedback [13] among others.

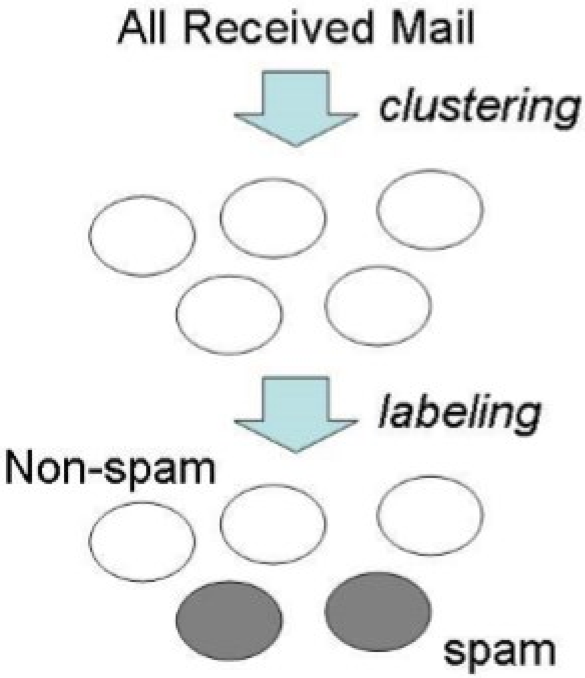
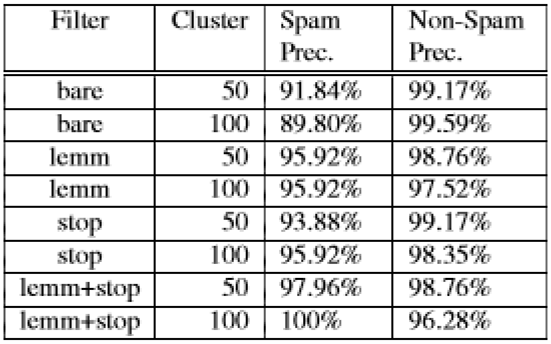


Figure 2. our spam detection model

3. Spam Detection System

Using the text clustering based on vector space model, we provide a novel spam detection method in this part. In this approach, the system finds spam more efficiently by automatically building the spam detection model using the contents of different types of mail. We utilize a clustering approach called the spherical-means algorithm[5] for all the received mail to build the spam detection model. The technique uses a predefined number of clusters to split the mail collection. The cluster centroid vectors are determined using the Cluster Representative method for every cluster. The clusters make it easy to calculate the similarity between fresh emails and the clusters. The spam content is shown as a one-term statistic in the previously suggested approaches, including the Naive Bayes classifier and the SVM filter. On the other hand, our technique uses centroid vectors to describe the contents of different types of mail as many term statistics. After acquiring the centroid vectors, the label ('spam' or 'non-spam') is determined by determining the cluster's spam mail count. We classify a cluster as spam if the proportion of spam messages relative to the total number of messages in the cluster exceeds the threshold of 70% to 85%. Partitioning a collection of clusters into spam and non-spam clusters is therefore possible. For each incoming email, the system determines if it is spam based on the centroidvectors of the spam and non-spam clusters. To begin, the vector in is applied to newly received messages in the same manner as the information retrieval paradigm based on vector spaces. When we have the vector in hand, we can compare it to the centroid vector for each cluster and determine their cosine similarity. The last step is to give the new email the label of the cluster that is most applicable.



Filter	Cluster	Spam Prec.	Non-Spam Prec.
bare	50	91.84%	99.17%
bare	100	89.80%	99.59%
lemm	50	95.92%	98.76%
lemm	100	95.92%	97.52%
stop	50	93.88%	99.17%
stop	100	95.92%	98.35%
lemm+stop	50	97.96%	98.76%
lemm+stop	100	100%	96.28%

Table 1. Experimental results of our system

4. Experiment

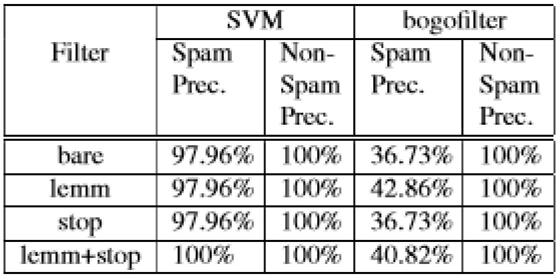
The components of a vector representing an email document are given two-part values. [14]

$$w_{ij} = L_{ij} \times G_i .$$

According to our findings, a factor may be either a local weight that represents the importance of a word within a document or a global weight that represents the total worth of a term as an indexing term for the whole set of documents.

$$w_{ij} = L_{ij} \times G_i = f_{ij} \cdot \log \frac{n}{df_i},$$

in where is the total number of documents in the collection, is the frequency of the -th phrase in the -th document, and is the total number of documents that include the -th term. We conduct experiments with spam detection using the publicly accessible LingSpam test collection in order to assess the efficiency of the four-system. The Ling-Spam collection includes 481 spam messages that were manually classified and 2412 nonspam communications. This collection is comprised of four components: bare, lemm, stop, and lemm+stop. The first three components are addressed independently by lemmatizer and stop-list, respectively. Our results show that there are 432 spam messages and 1,170 non-spam messages in the data set, with 242 non-spam messages and 49 spam messages in the test set. You can see the experimental findings in Table 1. Both spam and non-spam communications are handled by our system with good performance in this figure. The accuracy rate for spam is above 90%, while the accuracy rate for non-spam is over for every single collection, 96%. Furthermore, in order to conduct a fair assessment of our approach, we compare its accuracy with that of other methods. Support Vector Machine (SVM)[8] and bogofilter[3] are used in this comparison. Boofilter is a Bayesian spam filter, and support vector machines (SVMs) are among the most sophisticated machine learning methods. In table 2, we display the outcome. Compared to the bogofilter and almost on par with the SVM, our technique achieves superior accuracy, as seen by these results. It follows that an efficient strategy for spam detection is the use of unsupervised clustering based on spam and non-spam clusters. So that the non-spam precision is not 100%, we define the spam cluster threshold value as 70%. We determine the spam precision if the non-spam precision is close to 100% and the threshold value if it is larger than 70%. The outcomes of these trials using TF IDF (Text Frequency Inverse Document Frequency) and TF as term weights are shown in tables 3 and 4, respectively. Our technique delivers high-performance for spam messages, with a spam accuracy of roughly 90%. Another benefit of TF over TF IDF for spam detection is an improvement in accuracy, with the exception of the lemm stop result.

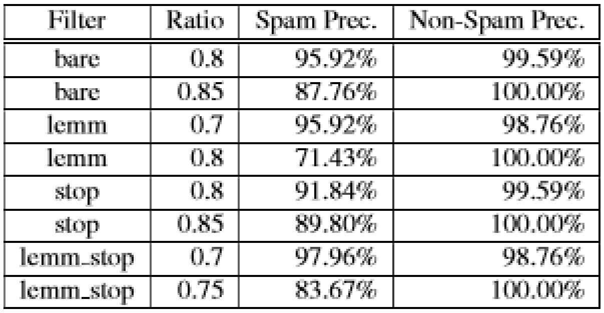


Filter	SVM		bogofilter	
	Spam Prec.	Non-Spam Prec.	Spam Prec.	Non-Spam Prec.
bare	97.96%	100%	36.73%	100%
lemm	97.96%	100%	42.86%	100%
stop	97.96%	100%	36.73%	100%
lemm+stop	100%	100%	40.82%	100%

Table 2. Experimental results using SVM and bogofilter

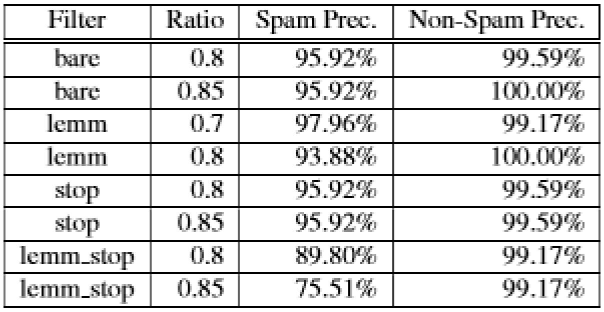
5. Conclusion

We provide a novel method for spam detection in this research that makes use of text clustering in a vector space model. Using the contents of different types of email, this method constructs the spam detection model and finds spam more efficiently. The experimental findings demonstrate that our solution outperforms the bogofilter and achieves accuracy close to that of the SVM. It follows that an efficient strategy for spam detection is the use of unsupervised clustering based on spam and non-spam clusters. To further refine the spam and non-spam clusters utilizing dynamic updates, like relevance feedback, further work is needed.



Filter	Ratio	Spam Prec.	Non-Spam Prec.
bare	0.8	95.92%	99.59%
bare	0.85	87.76%	100.00%
lemm	0.7	95.92%	98.76%
lemm	0.8	71.43%	100.00%
stop	0.8	91.84%	99.59%
stop	0.85	89.80%	100.00%
lemm_stop	0.7	97.96%	98.76%
lemm_stop	0.75	83.67%	100.00%

Table 3. Experimental results using some threshold values(TFIDF)



Filter	Ratio	Spam Prec.	Non-Spam Prec.
bare	0.8	95.92%	99.59%
bare	0.85	95.92%	100.00%
lemm	0.7	97.96%	99.17%
lemm	0.8	93.88%	100.00%
stop	0.8	95.92%	99.59%
stop	0.85	95.92%	99.59%
lemm_stop	0.8	89.80%	99.17%
lemm_stop	0.85	75.51%	99.17%

Table 4. Experimental results using some threshold values(TF)

References

[1] I. Androutsopoulos, J. Koutsias, K. Chandrinos, G. Paliouras, and C. Spyropoulos. An evaluation of naive bayesian anti-spam filtering. In Proceedings of the Workshop on Machine Learning in the New Information Age: 11th European Conference on Machine Learning (ECML 2000), pages 9–17, 2000.

[2] I. Androutsopoulos, G. Paliouras, V. Karkaletsis, G. Sakkis, C. Spyropoulos, and P. Stamatopoulos. Learning to filter spam e-mail: A comparison of a naive bayesian and a memory-based approach. In Proceedings of the workshop Machine Learning and Textual Information Access, 4th European Conference on Principlesand Practiceof Knowledge Discovery in Databases (PKDD-2000), pages 1–13, 2000.

[3] Bogofilter. http://bogofilter.sourceforge.net/.

[4] W. W. Cohen. Learning rules that classify e-mail. In Proceedings of the 1996 AAAI Spring Symposium on Machine Learning in Information Access, pages 203–214, 1996.

[5] I. S. Dhillon and D. S. Modha. Concept decompositions for largesparsetextdatausingclustering. Technicalreport,IBM Almaden Research Center, 1999.

[6] H. Druker. Support vector machines for spam categorization. In Proceedings of the IEEETransaction on Neural Networks, volume 10, pages 1048–1054, 1999.

[7] P. Graham. Better Bayesian Filtering. http://www.paulgraham.com/better.html.

[8] T. Joachims. Learning to Classify Text Using Support Vector Machines. Dissertation, Kluwer, 2002.

[9] A. Kolcz and J. Alspector. Svm-based filtering of e-mail spam with content-specific misclassification costs. In Proceedings of the TextDM!G01 Workshop on Text Mining, IEEE International Conference on Data Mining, pages 1048– 1054, 2001.

[10] J. R. Quinlan. C4.5: Programs for Machine Learning. Morgan Kaufmann, San Mateo, CA, 1993.

[11] M. Sahami, S. Dumais, D. Heckerman, and E. Horvitz. A bayesian approach tofilteringjunke-mail.

InProceedings of the AAAI-98 Workshop on Learning for Text Categorization, pages 1048–1054, 1998.

[12] G. Sakkis, I. Androutsopoulos, G. Paliouras, V. Karkaletsis, C. Spyropoulos, and P. Stamatopoulos. Stacking classifiers for anti-spam filtering of e-mail. In Proceedings of the 6th Conference onEmpiricalMethods inNaturalLanguage Processing(EMNLP 2001), pages 44–50, 2001.

[13] G. Salton and C. Buckley. Improving retrieval performance by relevance feedback. Journal of the American Society for Information Science, 41:288–297, 1990.

[14] I. H. Witten, A. Moffat, and T. C. Bell. Managing Gigabytes: Compressing and Indexing Documents and Images. Van Nostrand Reinhold, New York, 1994.