# Analysis on Different Classifiers for the Study of SMS Spam Filtering

**Gege Gao**
School of Information Studies
Syracuse University
Syracuse, NY 13210 USA
gegao@syr.edu

## Abstract

With the development of mobile phones, Short Message Service (SMS) is being used by most people in the world. Therefore, spam messages are dramatically growing as well. However, blocking these spam messages can be difficult. The main reasons include insufficient storage to install spam filter softwares on mobile phone and insufficient contents for spam filter to classify the spam. This paper uses mass SMS data to test several different classifiers on several different vectors for spam filtering, and reaches the conclusion that MNB has the best performance on SMS spam filtering which could be used in the future spam filtering.

## Keyword

Spam filtering, SMS, Classification

## 1 Introduction

Short Message Service (SMS) is a text messaging communication service on phone, web or other mobile devices. According to the Analysis Mason[1], about 17 trillion messages are sent via SMS, which means SMS is a prominent way for users to communicate in 2014. Thus, many spam messages sent to users' mobile phones just as the junk mail does. The report from Cloudmark[2] shows that from January to February in 2014, Twilio (a spammer company) has attacked over a quarter of a million US mobile phone subscribers, sending over 385,000 messages from about 2,500 unique phone numbers.

SMS spam is not only annoying but may cost a lot in certain countries. Therefore, we should do more studies to deal with it. However, it is not easy for researchers or field experts to work it out. Since the standard SMS messaging is limited to 140 bytes, many idioms and abbreviations are used in the message text, which is not easy to be recognized by computer.

This paper uses 5573 raw messages, with ground truth as spam or ham from exist data set[3] collected by Tiago, José and Akebo. (Almeida, Hidalgo & Yamakami, 2011) This data set includes messages manually extracted from the Grumbletext Website[4], a subset of SMS randomly chosen ham (non-spam) messages of NUS SMS Corpus[5], and SMS messages collected from Caroline Tag's PhD Thesis[6] and the SMS Spam Corpus v.0.1Big[7]. Table 1 shows the composition of this data set.

Table 1 data set composition

| Message | Amounts | Percentage |
|---|---|---|
| Hams | 4826 | 86.60% |
| Spams | 747 | 13.40% |
| Total | 5573 | 100.00% |

[1]See: http://www.analysysmason.com/Research/Content/Reports/Mobile-digital-economy-Feb2014/Mobile-and-the-digital-economy/
[2]See: http://blog.cloudmark.com/2014/02/13/sms-phishers-exploit-twilio-and-owly-to-steal-mobile-account-logins/
[3] This data set is public available at http://www.dt.fee.unicamp.br/~tiago/smsspamcollection/
[4] See: http://www.grumbletext.co.uk/
[5]See:http://www.comp.nus.edu.sg/~rpnlpir/downloads/corpora/smsCorpus/
[6] See: http://etheses.bham.ac.uk/253/1/Tagg09PhD.pdf
[7] See: http://www.esp.uem.es/jmgomez/smsspamcorpus/

## 2 Method

To make the analysis more accurate, the data set needs cleaning. Tiago, José and Akebo have done the duplication analysis for this data set. Then they use different tokenizers to eliminate useless content. In this paper, I manually remove the useless symbols such as ":)", ":(" and apostrophe.

For evaluating the performances of different classifiers, I choose the Expectation-Maximization (EM) clustering algorithm as the baseline. EM is an iterative method of finding the maximum likelihood estimate of the parameters from the dataset and I think it could be the baseline, for it could cluster different texts. In this paper, I use the default values except setting up the maximum iterations as 20 and random seed as 50 to cluster the data using Weka[8].

Table 2 is a list of classifiers used in this paper. Most of the classifiers are conducted with default values in Weka. (Witten & Frank, 2005) The Multinomial Naïve Bayes (MNB) classifier and Support Vector Machine (SVM) classifier are conducted using Scikit-learn[9] toolkit. (Pang, Lee & Vaithyanathan, 2002) The values are set default, except the minimum word term frequency=5 in all classifiers and parameter C=0.5 in SVM classifier. (Forman, Scholz & Rajaram, 2009) Additionally, different vectors like Boolean, Term frequency and Tf*idf as well as Unigram, Bigram and Trigram are used to get the best performances on MNB and SVM.

Table 2 Classifiers

| Classifiers |
|---|
| MNB |
| Linear SVM |
| Logistic |
| PART |
| JRip |
| J48 |
| Ibk |
| Ib1 |
| EM (baseline) |

Tiago, José and Akebo split first 30% of the data as training data and the remainder as testing data in their paper. I choose the first 3999 data as training data, and the last 1574 data as test data. Then I use 5-fold cross validation on training data to train the classifiers in order to predict the spam or ham based on testing data. Finally I compare the prediction with the ground truth and evaluate the classifier performance.

## 3 Result

### SVM and MNB evaluation on training data

To evaluate the performances of different vectors on SVM and MNB, I use the accuracy as the evaluation method. Table 3 and Table 4 demonstrate the performances of SVM and MNB.

Table 3 Different vectors in SVM

| Vector | Accuracy |
|---|---|
| Uni+Bi+Tri+Term frequency | 0.979744368 |
| Unigram+Term frequency | 0.979744055 |
| Unigram+Boolean | 0.979244681 |
| Uni+Bi+Boolean | 0.978744055 |
| Uni+Bi+Tri+Boolean | 0.978493742 |
| Uni+Bi+Term frequency | 0.978244368 |
| Uni+Bi+Tri+Tf*idf | 0.972992804 |
| Uni+Bi+Tf*idf | 0.972992491 |
| Uni+Tf*idf | 0.972242491 |

Table 4 Different vectors in MNB

| Vector | Accuracy |
|---|---|
| Uni+Bi+Tri+Boolean | 0.980495307 |
| Unigram+Bi+Boolean | 0.980244994 |
| Uni+Bi+Tri+Term frequency | 0.979744368 |
| Uni+Bi+Tf*idf | 0.979494055 |
| Unigram+Boolean | 0.979244994 |
| Unigram+Term frequency | 0.979244681 |
| Uni+Tf*idf | 0.972743429 |
| Uni+Bi+Tf*idf | 0.972243429 |
| Uni+Bi+Tri+Tf*idf | 0.971242804 |

As can be seen from the above, only slightly differences occur to the results of different vectors. Finally, Unigram, Bigram, Trigram and Term frequency vectors has the best performance in SVM classifier and get the accuracy of 0.979744368; Unigram, Bigram, Trigram and Boolean vectors has the best performance in MNB classifier and get the accuracy of 0.980495307. These two numbers will represent

[8]Most classifiers are provided by Weka, available at http://www.cs.waikato.ac.nz/ml/weka/

[9] Scikit-learn toolkit is based on Python, available at http://scikit-learn.org/stable/

the SVM and MNB to compare with other classifiers.

**Classifier Evaluation on training data**

Table 5 shows the accuracies on training data using different classifiers. From below, we can see that MNB has the best performance compared to other classifiers on training data.

Table 5 Classifier performance on training data

| Classifiers | Accuracy |
|---|---|
| MNB | 98.05% |
| Linear SVM | 97.97% |
| Logistic | 96.40% |
| PART | 95.77% |
| JRip | 95.62% |
| J48 | 95.27% |
| Ibk | 94.42% |
| Ib1 | 94.42% |
| EM (baseline) | 71.84% |

**Classifier Evaluation on testing data**

Tiago, José and Akebo employ the well-known performance measures: Spam Caught (*SC%*), Blocked Hams (*BH%*), Accuracy (*Acc%*), and Matthews Correlation Coefficient (*MCC*) to evaluate the classifiers. Since the trend of MCC is the same as SC and ACC, I use SC, BH, and ACC to make the performance measurement. Table 6 shows the final results of classifiers on testing data. (Almeida, Yamakami & Almeida, 2009)

Table 6 Classifier performance on testing data

| Classifiers | *SC%* | *BH%* | *Acc%* |
|---|---|---|---|
| MNB | **90.61%** | 0.73% | **98.10%** |
| Linear SVM | 88.26% | **0.51%** | 97.97% |
| JRip | 84.98% | 1.84% | 96.38% |
| PART | 85.45% | 2.35% | 96.00% |
| J48 | 78.40% | 1.54% | 95.74% |
| Logistic | 84.04% | 6.25% | 92.44% |
| Ibk | 63.85% | 13.08% | 83.80% |
| Ib1 | 63.85% | 13.15% | 83.74% |
| EM (baseline) | 82.63% | 9.18% | 89.71% |

As we can see in the table above, MNB has the best performance on the testing data with the highest accuracy at 98.10% and highest spam caught at 90.61%. However, Linear SVM has the lowest blocked hams rate at 0.51%. According to the result of Tiago, José and Akebo, the best performance is SVM with spam caught rate at 83.10% and accuracy at 97.64%, which are worse than the best result in this paper.

It is worth mentioning that JRip and PART classifiers also have good performances on testing data since they have a balance on spam caught and blocked hams. However, J48 classifier has a good performance on blocking ham but has a relatively lower result on spam caught. The Logistic classifier is opposite to J48. It has a good performance on spam caught but not so good a result on blocking ham. Ibk and Ib1 have relatively bad results on both spam caught and blocked ham.

Comparing the results of training data and testing data, we can see that MNB and Linear SVM perform well on both data. JRip, PART and J48 have the similar performance on both data. Logistic has a good performance on training data but a bad performance on testing data. EM baseline has the worst performance on training data but better than Ibk and Ib1 on testing data.

Overall, almost all the classifiers used in this paper have a better performance on testing data than baseline EM algorithm. However, Ibk and Ib1 classifiers perform worse than baseline on all variables. That is to say Ibk and Ib1 do not perform well on SMS spam filtering.

## 4 Conclusion

This paper aims at studying different classifiers on SMS spam filtering to see which classifier has a better performance. I use the existing large SMS message data set from Tiago, José and Akebo, which includes several different resources. Then I manually clean the data set, then I use cross validation to train the classifiers. By comparing these classifiers using different vectors on the data set provided by Tiago, José and Akebo, I get the result that MNB has the best performance both on training data and testing data; Linear SVM, though worse than MNB overall, has pretty good performance on avoiding blocking hams; either Ibk and Ib1 are worse than EM baseline, which is different from what the Tiago, José and Akebo get in their paper.

From my perspective, the reasons why I get different results from Tiago, José and Akebo are two. First, I use different vectors and different variable values from them. They use tokenizers, Boolean and Term frequency to classify, while I use Ngrams, Boolean, Term frequency and Tf*idf. They don't set the minimum word frequency while I set it to 5. Second, Tiago, José and Akebo have big testing data and small training data, while I have big training data and small testing data. What's more, I use 5-fold cross validation to

run the training data. Therefore, I have a better performance result.

Future work could focus more on the abbreviation of the ham text and the collection and separation of the data set as well as the algorithms on classification such as decision tree, Boosting and Bagging.

## 5 Acknowledgements

I would like to thank Professor Bei Yu for teaching me text mining and giving me instruction on this paper.

## Reference

Almeida, T., Yamakami, A., & Almeida, J. (2009, December). Evaluation of approaches for dimensionality reduction applied with naive bayes anti-spam filters. In Machine Learning and Applications, 2009. ICMLA'09. International Conference on (pp. 517-522). IEEE.

Almeida, T. A., Hidalgo, J. M. G., & Yamakami, A. (2011, September). Contributions to the study of SMS spam filtering: new collection and results. In Proceedings of the 11th ACM symposium on Document engineering (pp. 259-262). ACM.

Forman, G., Scholz, M., & Rajaram, S. (2009, June). Feature shaping for linear SVM classifiers. In Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining (pp. 299-308). ACM.

Hsu, C. W., Chang, C. C., & Lin, C. J. (2003). A practical guide to support vector classification.

Pang, B., Lee, L., & Vaithyanathan, S. (2002, July). Thumbs up? : sentiment classification using machine learning techniques. In Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10 (pp. 79-86). Association for Computational Linguistics.

Witten, I. H., & Frank, E. (2005). Data Mining: Practical machine learning tools and techniques. Morgan Kaufmann.