IST 736

Text Mining

Professor Yu *Yunteng Gu, Nan Sun, Jiarong Yang*

**SMS CLASSIFICATION FOR SPAM DETECTION**

1. **Introduction**

In modern society, short messages have become the inseparable part of people’s lives. However, a large number of illegal messages grow rapidly. Increasing spam messages both affect users’ work and life and add the load of the network, resulting in more resources mobile telecom carriers have to spend much time on filtering messages. Additionally, SMS spams are particularly more irritating than email spams, since sometimes they contribute to a cost for the receiver as well. These factors along with the limited availability of mobile phone spam-filtering software make it become an interesting topic to dig. Therefore in this project, we intend to apply different text mining and natural language processing knowledge to this problem and compare their performance to design an application on classifier of spam messages.

1. **Dataset Description**

The dataset is downloaded from Kaggle website which contains 3000 original text messages. All of them have been tagged as ‘spam’ or ‘ham’. In this dataset, the number of spam messages is 390, occupying 13% and the number of ham messages is 2610, occupying 87%. Therefore, our baseline is 87%. Since the majority vote baseline is too high to follow, we will also use precision, recall, and F-measure to evaluate our classification models.



Figure 1. Common words in the messages

1. **Model Design**

After reviewing our dataset, we find out that each record of messages is tagged as either spam or ham. What we need is a classifier to identify the character of one message. After our research in different models, we selected two kinds of which could be utilized to build our classifier, MultinomialNaiveBayes(MNB) and Support Vector Machine(SVM). Since MNB is super simple, runs more quickly and needs less training data while SVM has higher accuracy and excellent guarantees regarding overfitting.

In reality, messages are classified by whether senders are from unknown numbers and time messages sent. However, one critical assumption on our models is that we only focus on the content of messages. In other words, we will classify the spam messages by analyzing linguistic features of messages.

The first thing we need to do is to build MNB and SVM models only with unigram. We set the minimum document frequency as 5.

For the MNB model, we used both Sklearn and NLTK packages to create Naive Bayes model. Keeping all parameters unchanged, we get the results as following:

|  |  |  |
| --- | --- | --- |
| Sklearn | Ham (predict) | Spam (predict) |
| Ham (real) | **514** | **11** |
| Spam (real) | **2** | **73** |

|  |  |  |
| --- | --- | --- |
| NLTK | Ham (predict) | Spam (predict) |
| Ham (real) | **460** | **64** |
| Spam (real) | **0** | **76** |

Table 1. The Confusion Matrix of MNB model for SKlearn and NLTK package

|  |  |  |
| --- | --- | --- |
| MNB model | Sklearn | NLTK |
| Overall Accuracy: | 0.97833333 | 0.8933333 |
| ham precision: | 0.9961 | 1 |
| ham recall: | 0.979 | 0.8778626 |
| ham F-measure: | 0.9875 | 0.9349593 |

Table 2. The results for MNB models

For the SVM model, we used same two different packages to create SVM model. Meanwhile, for comparison, we tuned the model by changing the cost of SVM models. Here are the results of accuracy with the different cost:

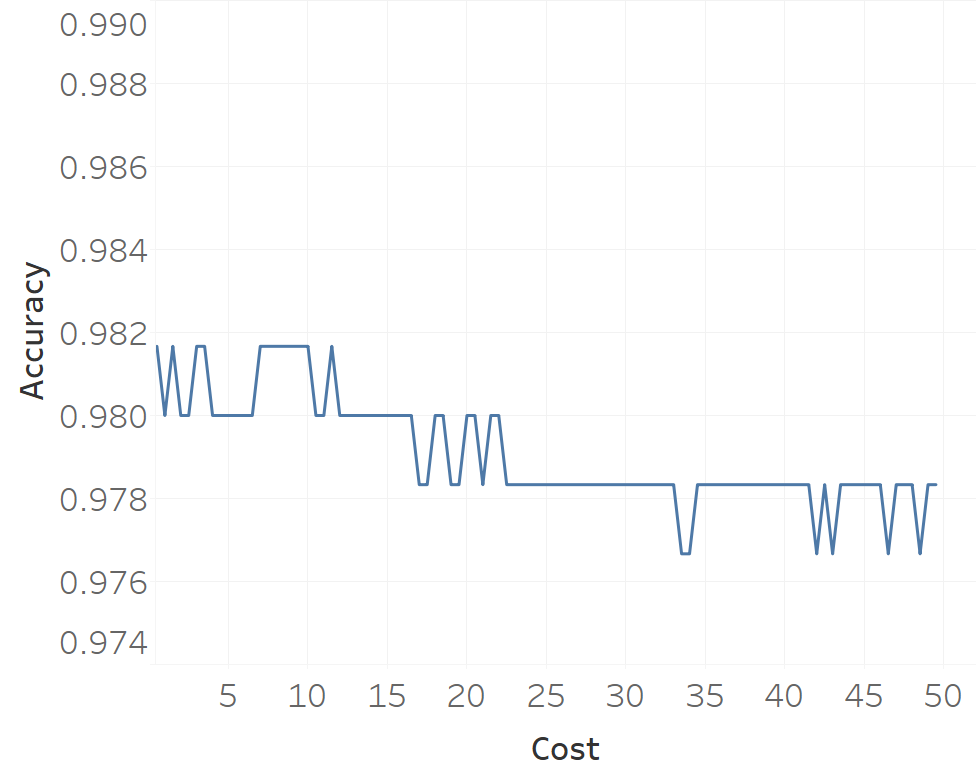


Figure 2. The accuracy with different cost for SVM.

After tuning the SVM model, we found that it shows the better accuracy when the cost is about 8. Thus we used the SVM model with cost equaling to 8 and got the results here:

|  |  |  |
| --- | --- | --- |
| Sklearn | Ham (predict) | Spam (predict) |
| Ham (real) | **522** | **3** |
| Spam (real) | **8** | **67** |

|  |  |  |
| --- | --- | --- |
| NLTK | Ham (predict) | Spam (predict) |
| Ham (real) | **524** | **0** |
| Spam (real) | **11** | **65** |

Table 3. The Confusion Matrix of SVM model for SKlearn and NLTK package

|  |  |  |
| --- | --- | --- |
| SVM model | Sklearn | NLTK |
| Overall Accuracy: | 0.98166667 | 0.9816667 |
| ham precision: | 0.9849 | 0.9794393 |
| ham recall: | 0.9943 | 1 |
| ham F-measure: | 0.9896 | 0.9896128 |

Table 4. The results for SVM models

According to the results above, the SVM models shows the better results on the spam message classification problem. Also compared two different Python packages, the Sklearn provided the higher performance result. Therefore, in the further discussion, we will only use Sklearn package to create and tuning models.

1. **Model Optimizing**

**4.1 Experiment I:**

Stop words can be an important factor we need to take into consideration. So in this experiment, we removed stop words in our dataset and run our previous models and tried to figure out whether the stop words effect on the performance. The results were shown following:

|  |  |  |
| --- | --- | --- |
|  | MNB | SVM |
| Overall Accuracy: | 0.978333333333 | 0.985 |
| ham precision: | 0.9961 | 0.99 |
| ham recall: | 0.9790 | 0.99 |
| ham F-measure: | 0.9875 | 0.99 |

Table 5. The results for Experiment I – removing stop words

The accuracy increased compared with our basic model. It turns out that removing stop words took effect. Therefore, we would continue removing stop words to finish following experiments.

**4.2 Experiment II**

Also, we would like to check whether using N-gram would help improve our model performance. We used both bigram and trigram integrated into our models. And as the same what we did above, we run the models again after removing stop words.

Using bigram: Using trigram:

|  |  |  |
| --- | --- | --- |
|  | MNB | SVM |
| Overall Accuracy: | 0.981666667 | 0.985 |
| ham precision: | 0.99 | 0.99 |
| ham recall: | 0.99 | 0.99 |
| ham F-measure: | 0.99 | 0.99 |

|  |  |  |
| --- | --- | --- |
|  | MNB | SVM |
| Overall Accuracy: | 0.981666667 | 0.985 |
| ham precision: | 0.99 | 0.99 |
| ham recall: | 0.99 | 0.99 |
| ham F-measure: | 0.99 | 0.99 |

Table 6. The results for Experiment II – Ngrams

**4.3 Experiment III**

During our process to build models, we found that there are some words appearing frequently in spam messages. For instance, phone number, URL address, money amount and email address. To improve our suppose, we would like to see the frequency of those words. So in the next step, we used regular expression to replace those strings. The patterns as following:

Phone number: “\d{5}\d+” or ”\d{2}\d+-\d\d\d+” or …=> “phonephonephone”

URL Address:” w{3}?\.[\S]\*\.com” or “http://.\*” or …=> “urlurlurl”   
 Amount of money: “\$\d?[,\.]?\d+ “or “ £\d?[,\.]?\d+”=>“moneymoneymoney”

Email Address: “.\*?@.\*?\..\*” or “.\*?@.\*?\..\*?\..\*” => “emailemailemail”

After that, we generated a word cloud graph’s for spam messages.

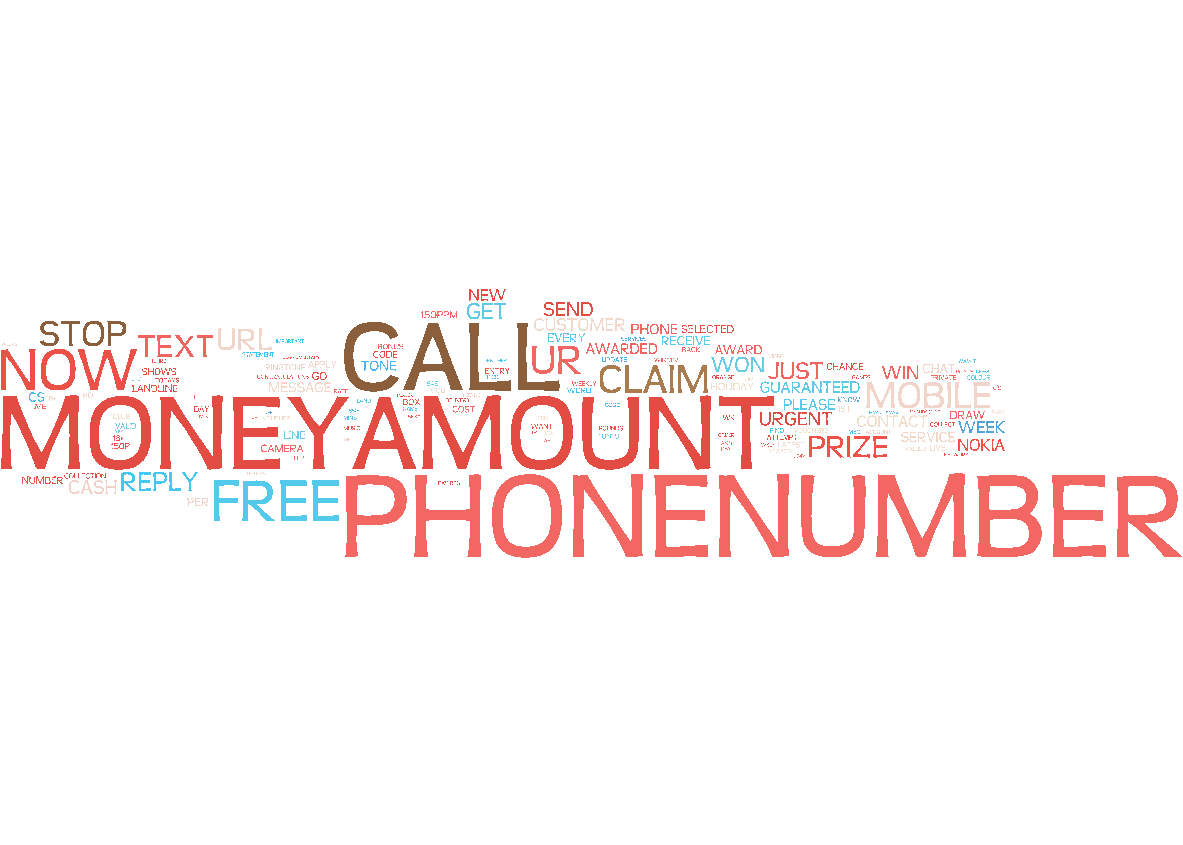


Figure 3. Common words in the spam messages

From word cloud graphs and scatter plots we drew, it turns out that phone number, URL address, money amount and email address appeared frequently.

Also, we conducted scatter plots for these four features.

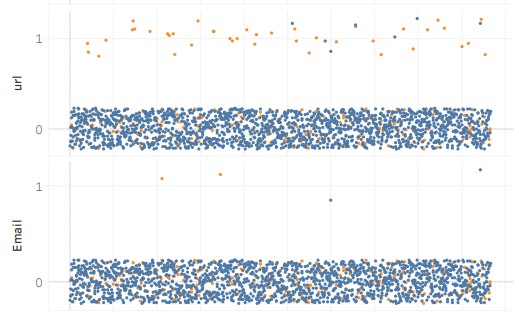
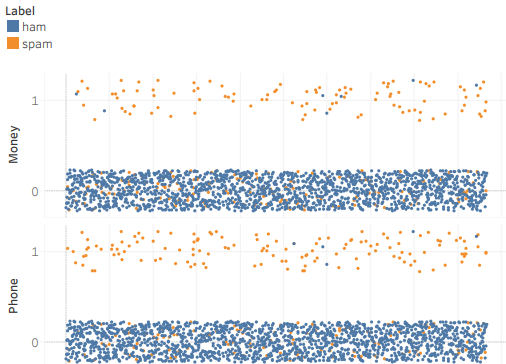


Figure 4. Scatter plots for the new four features

In the scatter plots, it shows most messages, which contains a phone number, URL, and amount of money, are most likely to be spam messages.

We decided to add these new words as new features and rerun our models. In the following experiment, we would build a model by using stepwise way. In other words, we would add one feature each time to find whether our accuracy would change. We first just used the phone number as one feature to test the results.

**Model 1**

We added the “phone number” as a new feature into the model from experiment II as model 1. The results show the messages with “Phone number” do help the prediction performance.

|  |  |  |
| --- | --- | --- |
|  | Ham (predict) | Spam (predict) |
| Ham (real) | **524** | **1** |
| Spam (real) | **4** | **71** |

|  |  |
| --- | --- |
|  | SVM model 1 |
| Overall Accuracy: | 0.9917 |
| ham precision: | 0.9924 |
| ham recall: | 0.9981 |
| ham F-measure: | 0.9953 |

Table 7. The results of advanced SVM model 1 with “Phone number” feature.

**Model 2**

Based on the model 1, we added “money amount” as a new feature into the model as Model 2. The results show the messages with “Money amount” do help the prediction performance.

|  |  |  |
| --- | --- | --- |
|  | Ham (predict) | Spam (predict) |
| Ham (real) | **524** | **1** |
| Spam (real) | **3** | **72** |

|  |  |
| --- | --- |
|  | SVM model 1 |
| Overall Accuracy: | 0.9933 |
| ham precision: | 0.9943 |
| ham recall: | 0.9981 |
| ham F-measure: | 0.9962 |

Table 8. The results of advanced SVM model 2 with Phone+Money

**Model 3**

Then we added “URL address” features into the Model 2 as Model 3.

|  |  |  |
| --- | --- | --- |
| Sklearn | Ham (predict) | Spam (predict) |
| Ham (real) | **523** | **2** |
| Spam (real) | **3** | **72** |

|  |  |
| --- | --- |
|  | SVM model 1 |
| Overall Accuracy: | 0.9917 |
| ham precision: | 0.9943 |
| ham recall: | 0.9962 |
| ham F-measure: | 0.9952 |

Table 9. The results of advanced SVM model 3 with Phone+Money+URL

In Model 3, the accuracy rates of SVM and MNB models went worse, which means the URL address didn’t affect our model. Therefore, it is not necessary to add this feature into our model.

**Model 4**

Then we added email address feature into Model 2 and run SVM model. However, like the model 3, the results went worse and we decided to remove this feature.

|  |  |  |
| --- | --- | --- |
|  | Ham (predict) | Spam (predict) |
| Ham (real) | **523** | **2** |
| Spam (real) | **3** | **72** |

|  |  |
| --- | --- |
|  | SVM model 1 |
| Overall Accuracy: | 0.9917 |
| ham precision: | 0.9943 |
| ham recall: | 0.9962 |
| ham F-measure: | 0.9952 |

Table 10. The results of advanced SVM model 3 with Phone+Money+email

**Model 5**

For our model 2, we used money and phone number as the features and its overall accuracy is the best among the others. Then we changed parameter settings to make the recall value better. Since our data set is skewed, we set class weight as 0.87 : 0.13 to balance the training set. And then We got our best model. Shown as following

|  |  |  |
| --- | --- | --- |
|  | Ham (predict) | Spam (predict) |
| Ham (real) | **525** | **0** |
| Spam (real) | **6** | **69** |

|  |  |
| --- | --- |
|  | SVM model 1 |
| Overall Accuracy: | 0.99 |
| ham precision: | 0.9987 |
| ham recall: | 1 |
| ham F-measure: | 0.9943 |

Table 11. The results of advanced weighted SVM model 5 with Phone+Money

In the Model 5, the accuracy of SVM decreased compared with the models. However, we preferred this model because our point is on the recall score.

**Conclusion**

According to our previous experiment, we found out that the optimal model is Model 2 in experiment III, which vectorized the text using unigram and removed all stop words from the dataset, with two extra features added, phone number and money amount.

Since we are working on a spam messages classification model, we should not only consider the precision of each category but also consider the recall of ham messages. Because if the model falsely gives regards a ham message as a spam message, the user will miss a ham message, which is not tolerable. However, if the model falsely gives a ham category to a spam message, the only result is that the user has to read one more spam message and will delete the message by himself (human annotation). Since our model has pretty high precision on spam messages, most of the spam messages will be removed and therefore, the efficiency of work is promoted a lot and no ham message will be missed.

**Reference**

[1] Ghosh, S., Srijith, P. K., & Desarkar, M. S. (2017). Using social media for classifying actionable insights in disaster scenario. International Journal of Advances in Engineering Sciences and Applied Mathematics, 9(4), 224-237. 10.1007/s12572-017-0197-2  
[2] Ghosh, S., Ghosh, K.: Overview of the FIRE 2016 micro-blog track: information extraction from micro-blogs posted during disasters. In: Working notes of FIRE 2016-Forum for Information Retrieval Evaluation, Kolkata, India, December 7–10, 2016, CEUR Workshop Proceedings. CEUR-WS.org (2016)