Spam Email Detection Beyond Naive Bayes, Optional Project Report for E4525 Machine Learning

Jiachuan Bi
SEAS, IEOR Department
Columbia University
jb4360@columbia.edu

Abstract

Naive Bayes (NB hereafter) spam filtering[1] is a straight-forward, well-studied and powerful statistical technique of e-mail filtering. In order to incorporate as much as information as possible, we designed structural information model, subject model, message model and combined model to utilize different level of data from text, using TREC 2005 data set[2]. Further investigation were conducted to compare other statistical learning methods with NB filters, including the cutting-edged LSTM neural network with words embedding layer. The generalization ability of the best model was tested by TREC 2006 data set[3].

1 Feature Engineering

Skipping all the emails with encoding error in TREC 2005 data set, 85203 emails were used for training, validation and testing. We extracted three classes of features: structural information, subject information and message body information. The first one contained only four Boolean variable: `multipart`, `html`, `links`, `attachments`. For the latter two text features, we applied bag-of-word representation, including set, count and tfidf features from subject text and message body text.

The extraction of more high-level features, including the vector representation of words used for LSTM, will be discussed in Section 3.

2 NB Models

Follow the instruction, we splited preprocessed TREC 2005 data set into 15% test data and 85% train data. 5-fold validation was adopted for the validation process. The final results of different multinomial NB models are presented in Table 1 and 2. The best smoothing parameter α was selected from set $\{0.0001, 0.001, 0.1, 1, 10, 100, 1000\}$, taking 5-fold cross-validated AUC as selection criteria. The combined feature model took all the information into consideration to form a larger feature space for training, while inr the combined probability model we trained three NB models respectively, and then combined them with another multinomial NB classifier, based on three predicted conditional probability.

The results in Table 1 and 2 suggested that structural info model performs poorly for the spam detection task, with only 0.576 accuracy on test data set, slightly better than random guess, which indicates that the information from structural data is insufficient to detect spam.

So, how about the information from text? From Table 1 and 2, we learned that all the models with text information achieve more than 0.9 AUC and more than 0.9 accuracy on both validation and test data set. It seemed that spam detection is not a hard job even if we apply such a simple model (NB classifier with bag-of-words representation). The best NB model achieved 0.996 AUC and 0.976 accuracy on test set. If we only use subject information without message body (often much more longer than subject), we can still get highest AUC 0.986 and accuracy 0.927 on test data set.

Model/Features	Set	Count	Tfidf
Structural Info Model	0.722(0.761)	/	/
Subject Model	0.985(0.982)	0.985(0.981)	0.986(0.983)
Message Body Model	0.962(0.959)	0.960(0.957)	0.967(0.964)
Combined Feature Model	0.975(0.973)	0.972(0.970)	**0.996(0.996**)
Combined Probability Model	0.975(0.973)	0.972(0.970)	0.994(0.994)

Table 1: The Test (5-fold-Validated) AUC of Multinomial NB Models

Model/Features	Set	Count	Tfidf
Structural Info Model	0.576(0.572)	/	/
Subject Model	0.922(0.914)	0.921(0.914)	0.927(0.919)
Message Body Model	0.935(0.932)	0.921(0.917)	0.943(0.939)
Combined Feature Model	0.960(0.959)	0.960(0.957)	**0.976(0.975**)
Combined Probability Model	0.962(0.959)	0.958(0.956)	0.975(0.973)

Table 2: The Test (5-fold-Validated) Accuracy of Multinomial NB Models

What's more, we noticed that the test results were always better than the validation results. The reason is, as a generative model, NB has great generalization ability. Besides, when testing with new data, all of the training data were used for training, instead of using only 0.8 training data during 5-fold cross validation, so the better test result also thanked to more training data during the re-training and test stage.

3 Comparison with Other Models

Other statistical and deep learning models are great competitors of NB for the spam detection task. We tried several other ML models and listed results in Table 3 and 4. All the models took the locally best parameters, except for the simple RNN model and LSTM model (using the default settings). Since some models took lots of time to train (e.g. SVM and boosting-based model), due to the constraint of computing power, we used only subject information (in Section 2 we have seen that subject can provide enough information for spam detection) and 8500, 1500 samples for training and test.

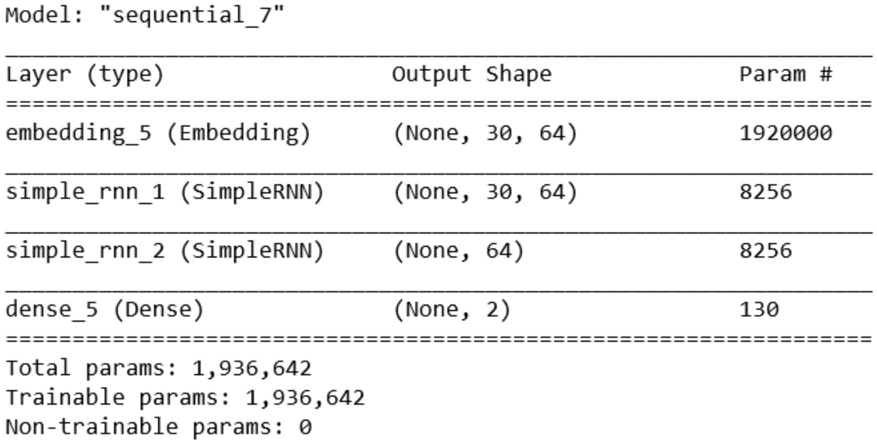
The features used for LSTM and RNN neutral network was sequences produced by an embedding layer before two hidden NN layers, with embedding dimension 64. During the preprocessing, we tokenized the email subjects by using first 30000 most frequent words, truncated the length of them to 30 words and padded with 0 for shorter sequences. The default structures for these two neutral network are shown in Figure 1 and 2, training with adam optimizer and batch size 100. Since it took a long time to train, we did not tune these structures, but they still worked pretty good.

Model/Features	Set	Count	Tfidf	Vector
Multinomial NB (Benchmark)	0.986(0.984)	0.987(0.984)	0.986(0.986)	/
KNN	0.938(0.920)	0.939(0.921)	0.924(0.895)	/
SVM with RBF kernel	0.982(0.976)	0.982(0.977)	0.976(0.972)	/
Logistics Regression	0.982(0.976)	0.982(0.977)	0.976(0.972)	/
Random Forest	0.989(0.986)	0.988(0.986)	**0.990(0.988**)	/
Xgboost	0.988(0.986)	0.989(0.986)	0.990(0.987)	/
Two Hidden Layers Simple RNN	/	/	/	0.979(0.979)
Two Hidden Layers LSTM	/	/	/	0.981(0.985)

Table 3: The Test (5-fold-Validated) AUC of different ML Methods, using 10000 samples' subject

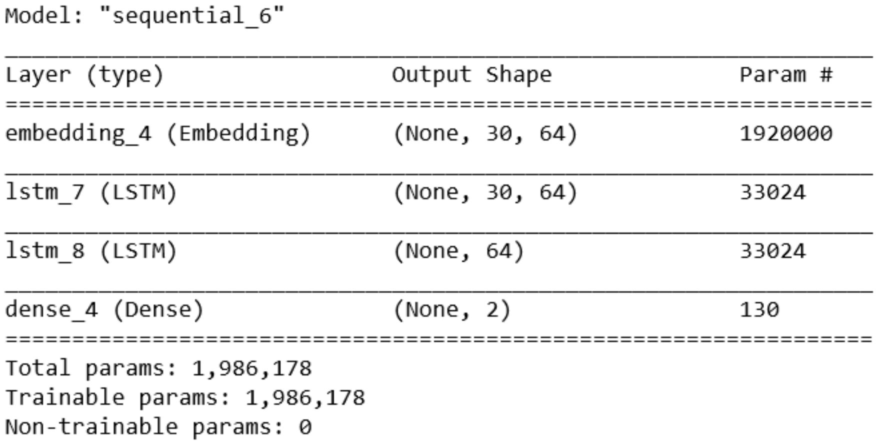
Model/Features	Set	Count	Tfidf	Vector
Multinomial NB (Benchmark)	0.948(0.936)	0.943(0.935)	0.941(0.943)	/
KNN	0.881(0.883)	0.879(0.884)	0.852(0.885)	/
SVM with RBF kernel	0.923(0.916)	0.924(0.916)	0.938(0.934)	/
Logistics Regression	0.923(0.916)	0.924(0.916)	0.938(0.934)	/
Random Forest	0.951(0.948)	0.952(0.947)	0.950(0.946)	/
Xgboost	0.951(0.947)	0.949(0.947)	0.950(0.946)	/
Two Hidden Layers Simple RNN	/	/	/	0.946(0.936)
Two Hidden Layers LSTM	/	/	/	**0.953(0.949)**

Table 4: The Test (5-fold-Validated) Accuracy of different ML Methods, using 10000 samples' subject



```
Model: "sequential_7"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
embedding_5 (Embedding)      (None, 30, 64)            1920000
_________________________________________________________________
simple_rnn_1 (SimpleRNN)     (None, 30, 64)            8256
_________________________________________________________________
simple_rnn_2 (SimpleRNN)     (None, 64)                8256
_________________________________________________________________
dense_5 (Dense)              (None, 2)                 130
=================================================================
Total params: 1,936,642
Trainable params: 1,936,642
Non-trainable params: 0
```

Figure 1: Default Structures of Simple RNN



```
Model: "sequential_6"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
embedding_4 (Embedding)      (None, 30, 64)            1920000
_________________________________________________________________
lstm_7 (LSTM)                (None, 30, 64)            33024
_________________________________________________________________
lstm_8 (LSTM)                (None, 64)                33024
_________________________________________________________________
dense_4 (Dense)              (None, 2)                 130
=================================================================
Total params: 1,986,178
Trainable params: 1,986,178
Non-trainable params: 0
```

Figure 2: Default Sructures of LSTM

The two layers LSTM performed best in terms of validation and test accuracy, while random forest (with tfidf features) achieved the highest in and out-of-sample AUC. We will take LSTM as the best model and compare it with the combined probability model in Section 2, with the TREC 2006 data set.

4 Best Model Analysis with TREC 2006

Using all the data of TREC 2005 to train a combined feature multinomial NB model, we apply it to TREC 2006 data set. The test results are shown in Table 5 and 6, and the ROC curves are shown in Figure 3. For LSTM, the learning process and ROC curve are show in Figure 4 and 5.

Model/Features	Set	Count	Tfidf	Vector
Multinomial NB (Combined Feature Model)	**0.966**	0.929	0.964	/
Two Hidden Layers LSTM	/	/	/	0.913

Table 5: The Test AUC of NB and LSTM on TREC 2006

Model/Features	Set	Count	Tfidf	Vector
Multinomial NB (Combined Feature Model)	**0.916**	0.877	0.894	/
Two Hidden Layers LSTM	/	/	/	0.866

Table 6: The Test Accuracy of NB and LSTM on TREC 2006

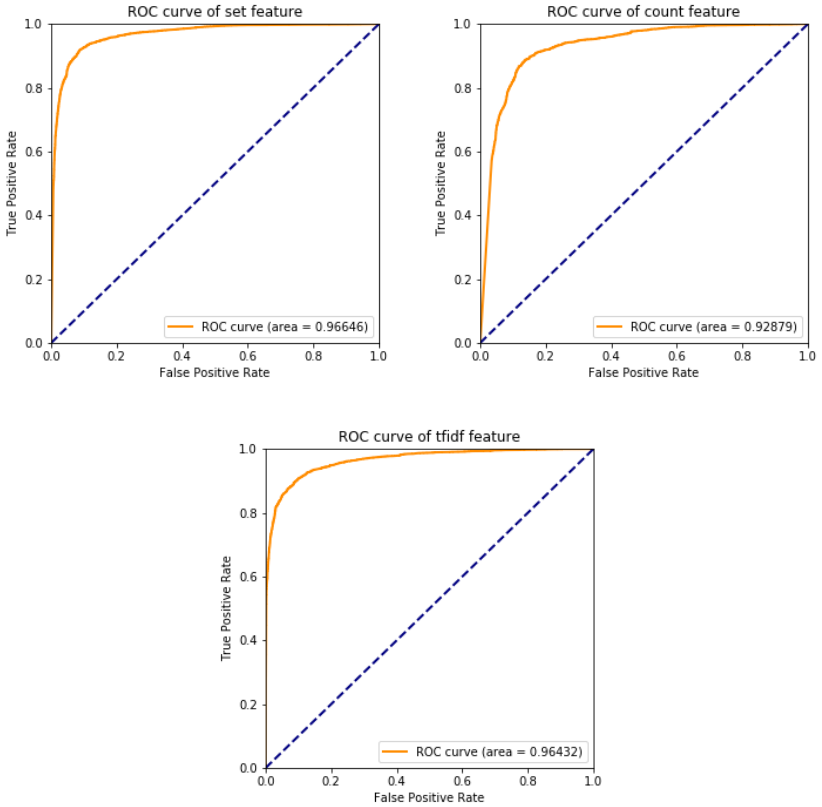


Figure 3: The ROC Curve of Multinomial NB (Combined Feature Model) with different bag-of-word feature

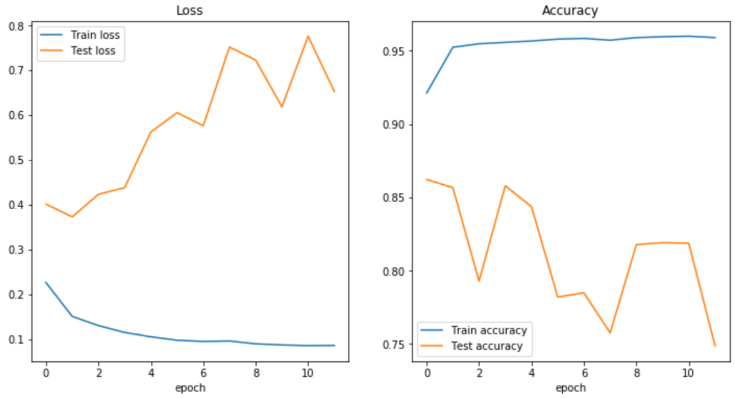


Figure 4: Learning Curve of LSTM

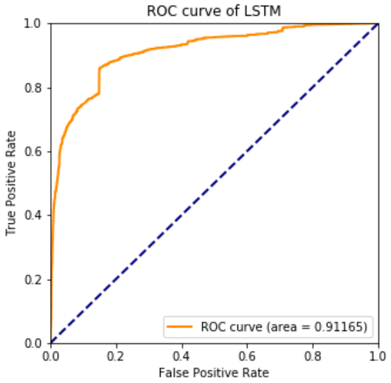


Figure 5: The ROC Curve of LSTM

5 Discussion

There are two main reasons contributing to the deviation between the two test results from TREC 2005 and TREC 2006. The first one is, they have different vocabulary sets. If one word appears in TREC 2006 and doesn't appear in TREC 2005, it will not be taken as effective information during tokenization process, which prevent effective detection. The second reason is, the spam emails in 2006 may develop some new ways to prohibit them from being detected, e.g., using some words appearing frequently in non-spam emails. These reasons make spam detection a harder job, which explains the worse test result on TREC 2006.

Most anti-spam systems allow users to correct the miss-classified emails. It helps system to get train data of higher quality, which contributes to better detection accuracy.

In Section , we observed that LSTM performed not so good on TREC 2006, while multinomial NB has better generalization ability on new test set. The reason is, the neutral network is a pure discriminant model, which has no assumption of generation process of data, while the NB is a generative model, pre-assuming the joint probability distribution, which will have stronger generalization ability if the assumptions are satisfied.

6 Appendix

All the source code of this project can be found on my github [1].

[1] `https://github.com/IVANBIJIACHUAN/Spam_detectation_with_ML`

References

[1] Naive Bayes Spam Filtering. https://en.wikipedia.org/wiki/Naive_Bayes_spam_filtering

[2] TREC05. https://plg.uwaterloo.ca/~gvcormac/treccorpus/

[3] TREC05. https://plg.uwaterloo.ca/~gvcormac/treccorpus06/