A Comparison of Modern Spam Detection Techniques on Text Messages

Yigao Li
Graduate Student of Analytics Program
Georgetown University
Washington, DC, United States
yl942@georgetown.edu

Abstract

Spam detection has become an important challenge for companies that provide communication services. The most basic spam detection algorithm is Naïve Bayes classifier, which is a baseline for other algorithms, and it can deal with limited training datasets. Modern data scientists have gone further and use many other machine learning or deep learning techniques on this task, such as Linear Discriminant Analysis and Recurrent Neural Network. We show that both performs better than baseline model with 4-8 percentages improvement.

Keywords: deep learning, feature selection, machine learning, spam detection, text messages

1 Introduction

The mobile phone market has experienced an exponential growth since the first appearance of smartphones. With the evolution of broadband cellular network technology, Short Message Service (SMS) can not only be provided merely by telecommunication companies, but can also transfer through Wi-Fi or cellular network. This allows cell phone users to send more information than traditional text messages with no more than 160 characters, and to save their money on text portion. As the popularity of SMS messages has increased since the first SMS message in 1992, the number of user-unwanted, commercial and harassing messages rises as well. Sometimes, SMS spam is more irritating than malicious emails. Attack messages can quickly fill in cell phone storages, causing waste of shared communication channels. Nowadays, since most smartphones have access to the Internet, fraud weblink in SMS messages can result in cell phone virus infection and even financial damages. In some rare cases, opening an SMS message can cost certain amounts of money. Therefore, spam detection is a crucial challenge for information security.

SMS spam detection is a text classification problem. Each text message is classified as either spam message or legitimate (often called “ham”). Many statistical and machine learning algorithms are used to detect spam messages, such as logistic regression, Bayes classifier, and support vector machines.

Naïve Bayes Classifier is a supervised machine learning classifier based on Bayes Theorem. It uses bag of words to find correlation between word tokens. By Bayes Theorem, it can calculate the probability that a message is spam or ham given each token in the sentence. It is a baseline model in this project because it gives a low false positive rate, which means there is less error predicting a legitimate message as spam.

Linear Discriminant Analysis is also a popular supervised machine learning method in pattern recognition, which finds the relationship between features and separates spam and legit messages by maximizing the distance between two labels and minimizing the variation within both spam and ham.

In the field of deep learning, the most useful artificial neural network to deal with time sequence

Table 1. List of words manually selected from dataset

account	answer	award	bonus	call	cash
claim	click	code	congrat	com	credit
guarantee	http	msg	net	password	pobox
private	prize	subscribe	text	txt	urgent
valid	wap	win	winner	won	www

Table 2. The classification results for NB1, NB2, LDA and RNN+LSTM

Training Set %	NB1	NB2	LDA	RNN+LSTM
20%	92.01%	95.15%	96.88%	
50%	90.56%	93.58%	96.70%	98.49%
80%	90.31%	94.17%	96.41%	

data is Recurrent Neural Network (RNN). Long-Short Term Memory (LSTM) is an extension for RNN. It extends memory in between experiences that have long lags from RNN's internal state.

2 Dataset and feature extraction

2.1 Dataset

The original SMS spam dataset was published on University of California, Irvine Machine Learning Repository. It has 747 spam messages and 4827 ham messages in text format. All spam messages are manually extracted and all ham messages are randomly chosen from 10,000 legitimate messages. Due to limited training sample for spam detection models in real life, which is obvious that huge number of new messages are sent every day, we will train all models with respect to various size of training set to examine their prediction accuracy.

2.2 Feature Extraction

To transfer string data to numeric ones, we convert these SMS messages to a matrix of token counts and use word count vector to represent each message.

As an additional step, we check through all spam messages particularly and manually find features from a normal person's point of view because spam messages are manually selected as mentioned in last section. We create new features such as the character length, length of the longest continuous combination of capital letters and numbers, the percentage of numbers and punctuations in a message except periods, commas and hyphens. Some word features are also selected from spam messages and there are total 30 words (Table 1) considered in our training models.

3 Methods

3.1 Naïve Bayes with Word Count Vector

Based on Bayes Theorem, the probability of a message being a spam is calculated by sum of probabilities of a message being a spam given counts of certain words. So, in this Naïve Bayes classifier, we convert data as a word count matrix (NB1), which is also our baseline model.

3.2 Naïve Bayes with manual features

Same baseline model as above, but here we apply Naïve Bayes classifier on our own features (NB2) introduced in Section 2.2.

3.3 Linear Discriminant Analysis

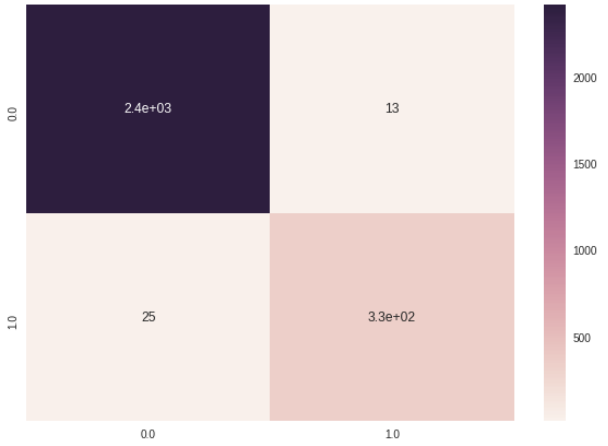
Linear Discriminant Analysis (LDA) extracts features from available attributes in order to separate messages of different labels, while grouping messages within each label as close as possible. In this project, we also apply this machine learning technique on features of our own.

3.4 Recurrent Neural Network

The only deep learning algorithm we test on this dataset is a Recurrent Neural Network in combination with Long-Short Term Memory (RNN+LSTM). RNN connects word tokens in a sequence, which allows exhibiting temporal dynamic behavior for a time sequence. It uses internal state that may have vanishing gradients while training. Thus, we use LSTM to keep the gradients steep. We expect the time spent for training can be short and high accuracy.

We consider only top 3800 words and make every message vector of dimension 380. For word embedding, we map each word onto a 32-length real valued vector. We pass each word vector to dropout layer of 0.2 to prevent overfitting, into each of 100 LSTMs, and then to sigmoid function which classifies a message to be either spam or ham.

Figure 1. RNN+LSTM prediction confusion matrix



4 Results and analysis

Overall, RNN+LSTM prediction has the highest accuracy with very high sensitivity. For machine learning techniques, LDA performs better than NB and human extracted features are more representative than word count vectors. At the same time, we should also see that Naïve Bayes classifier is good at dealing with small training dataset, and the size of training set does not influence LDA very much. (Table 2)

5 Conclusion

Several modern spam detection algorithms are introduced and trained on SMS spam dataset in this paper. Nowadays deep learning techniques are widely used in every field and some of them perform well on text classifications. Linear Discriminant Analysis, as a powerful machine learning feature extraction method, performs better than baseline Naïve Bayes classifier.

Acknowledgments

We are grateful to Professor Loehr and Professor Moreau for their excellent lectures on Natural Language Processing.

References

Giovane C. M. Moura, Anna Sperotto, Ramin Sadre, and Aiko Pras. 2013. Evaluating Third-Party Bad Neighborhood Blacklists for Spam Detection. Enschede, The Netherlands

Imani, Maryam and Montazer, Gholam Ali. 2017. Email Spam Detection Using Linear Discriminant Analysis Based on Clustering. 15. 22-30. Tehran, Iran

M. McCord and M. Chuah. 2011. Spam Detection on Twitter Using Traditional Classifiers. Bethlehem, PA

Sahil Puri, Dishant Gosain, Mehak Ahuja, Ishita Kathuria, Nishtha Jatana. 2013. Comparison and Analysis of Spam Detection Algorithms. New Delhi, India