Spam Detection

Amogh Dayal
dayal.a@husky.neu.edu

April 21, 2020

1 Motivation

Social media gives people an open platform to generate and consume large quantities of data. However, these sharing platforms are becoming increasingly infested with spam – repetitive, unrelated and unwanted messages. Kaspersky Lab reported that over half of the global email traffic between April 2019 and September 2019 was spam. [Vergelis, Shcherbakova, and Sidorina (2019)] Although email spam has decreased since 2012 [Clement (2019)], it may have simply moved to other platforms such as Twitter and Facebook which are increasingly threatened by "bots" that post spam and false information on these platforms. [*Study: Twitter bots played disproportionate role spreading misinformation during 2016 election* (2018)]

Furthermore, as new platforms emerge spam finds its way in. Thus, in order to effectively combat spam there is dire need for client-side spam detectors that can detect spam on multiple platforms. So far, spam detection efforts have focused on platform-specific spam and as of March 2020 there are no comparative studies that assess the different successful models on platform-independent datasets. In this project we implemented different spam detector models to compare them on a combined dataset derived from emails, Twitter and SMS text messages to identify the most robust model for spam detection across a diverse set of media.

2 Background

Spam detection is a modern instance of a binary classification problem. The task for the model is to classify a document - a message in a natural language of arbitrary length - as either "spam" or "not spam". The criteria for spam varies based on legal jurisdiction and media used for spamming; the broad definition that can be applied to spam is unsolicited bulk messages. Email spam typically serve as medium for sending promotional material, advertisements or malware. Spam on social media often posts irrelevant information while using the trending hashtags. While spam may use hyperlinks profusely on one platform, hyperlinks may be absent on another platform. On the other hand, profuse use of hyperlinks may be expected from legitimate users on a different platform. The variety in spam features on different platforms makes it difficult to classify spam using simple rule-based or signature-based systems and is also a significant challenge in creating a platform-independent spam filter.

Spam filters for email already exist that use Naive-Bayes for classification. [*Bogofilter* (2002)] While bogofilter was designed specifically for detecting email spam, the Naive-Bayes model can be applied for spam detection on other platforms as well. The Naive-Bayes model is often used as the baseline for measurement of performance in spam detection, but it's performance has been significantly improved upon by newer models such as k-means clustering and support vector machines.

Sasaki and Shinnou proposed a k-means clustering model for detecting spam. Centroids are calculated using spherical k-means algorithm and then label of "spam" or "not-spam" is applied to it based on what class the majority of the emails within that cluster fall under. They demonstrated that this form of spam detection performed significantly better than bogofilter with much higher true positive rate on the Ling-Spam dataset while giving comparable performance to a Support Vector Machines based model. [Sasaki and Shinnou (2005)] However, the performance of the model is unknown on messages from other media.

McCord and Chuah targeted spam on Twitter and compared the efficacy of Naive Bayes, K-nearest neighbors, Random Forest and Support Vector Machine models. They used features independent of the content of the tweet using the poster's past 100 tweets, followers and following to further improve spam detection based on poster history. They found that Random Forest model outperformed all others in precision, recall and f1 score. [Mccord and Chuah (2011)] However, on other social media platforms it is often difficult to gather such user-specific information which can have significant impact on their performance.

All of the studies mentioned earlier compared performance of different models on spam found on a single platform. As far as we know no studies have compared performance of spam detectors across multiple platforms. In light of increasing diversity in social media and means of communication, the ability of conventional machine learning techniques across multiple platforms is of great interest. In this project we compare the performance of successful models on a dataset derived from multiple sources of spam, in order to find the best performing model in different conditions for spam detection.

3 Methodology

The dataset used is composed of hand-labelled messages from SMS Spam Collection [*SMS Spam Collection v. 1* (n.d.)], tweets labelled by Chen, Yeo, Lau and Lee [Chen, Yeo, Lau, and Lee (2017)] and the Ling Spam dataset [*Ling Spam Dataset* (n.d.)]. The SMS Spam Collection is added to increase variety in the platforms for source of spam. All data was pre-processed to include only natural language content and all tags, html attributes and other metadata was stripped. All models were then evaluated using three mixtures of training and testing data. For each mixture of training and testing data the performance of the model was calculated as spam recall, spam precision, ham recall and ham precision. These four metrics offset the biased nature of the data where spam is less prevalent than

ham messages by capturing it's effect on the model's performance. A model with high bias to mark messages as ham would have a significantly low spam recall and vice versa. A model that correctly identifies spam but produces a lot of false positives would suffer penalties in ham recall. Thus the four metrics chosen give a holistic view of the model's overall performance.

3.1 Model Evaluation

In the first set of evaluations, the training and testing data was derived from the same dataset. The data was randomly divided into ten chunks of equal size to preform ten-fold cross validation. One of the chunks was marked as test data, and the rest of the chunks were used to train the model. Once training was complete, the model was used to predict classes for the test data samples and it's performance was evaluated as spam recall, spam precision, ham recall and ham precision. This process was repeated ten times; once for each chunk to be marked as test data. The measures were averaged over the ten trials. This analysis can be used to test the performance of each model on platform-homogeneous data, allowing identification of the best model for identifying spam on a single platform.

In the second set of evaluations, the training data composed of samples from two of the three platforms while the data from the remaining platform acted as the test set. The model was trained on the training data and then used to predict classes for the test dataset. The four metrics were then calculated and reported. This process was repeated three times; once for each platform's data marked as test data. This analysis can be used to test the performance of each model on previously unseen form of spam, simulating the emergence of a new source of spam as a new platform.

In the third set of evaluations, the training data was composed of a mixture of all datasets for all platforms. This data was then randomly divided into ten chunks of equal size for cross-validation. The model was trained on nine of the chunks while one chunk was marked as test data. The performance of the model was then calculated on the held-out test data after training. The four metrics were calculated based on the model's performance. This process was repeated for each chunk to be marked as test data. The average performance across the ten evaluations was reported. This analysis can be used to test the performance of each model on a mixture of messages from familiar but different platforms; allowing identification of the best model for a diverse set of platforms for which plenty of labelled messages already exist.

3.2 Models Tested

Four different models were implemented for testing: K-Means++ Clustering, Random Forest and Artificial Neural Networks. See addendum for the Python implementations of these models and the code used for evaluation.

3.2.1 K-Means++ Clustering

Each message was represented as a vector using the method proposed by Sasaki and Shinnou, where the elements of vector d are computed as

$$w_i = f_i \times \log \frac{n}{F_i}$$

where n is the number of messages, f_i is the frequency of term i in the current message and F_i is the number of messages in the training data that contain the term i [Sasaki and Shinnou (2005)]. Once the training data is transformed into a matrix formed of these vectors, PCA was used to reduce the dimensionality of the vectors to 50 elements. These vectors are then used to form 50 or 100 clusters. The class for each cluster is identified based on the class that the majority of messages in that cluster belong to.

3.2.2 Neural Networks

Each message was represented as a vector using the same method used for k-means clustering with PCA to reduce size of each vector to 50 elements. The classes were mapped to $Y = \{1, -1\}$ to represent spam and ham respectively. An artificial neural network was then created with hyperbolic tangent activation function and different sizes of hidden layers to be trained and evaluated. Results were reported for each tested combination of hidden layer sizes.

3.2.3 Random Forest

Each message was represented as a vector using the same method as previously described with PCA to reduce dimensions of each vector to 50. The random forest was then trained and evaluated. Results were reported for forests with different numbers of trees.

4 Results

4.1 Spam Detection On A Single Platform

The performance of the models on individual datasets is presented in Tables 1.1, 1.2 and 1.3 for SMS Spam, Ling Spam and Twitter Spam datasets respectively. In this evaluation, Random Forest with 50 trees was the best performing model, outperforming all other models on almost all four measures on all three datasets consistently. However, K-Means Clustering models with 50 centroids had the highest ham precision, indicating that the number of false positives for spam was the lowest for models based on K-Means Clustering with 50 centroids. Of the neural networks tested, models with 20 perceptrons in the first and second hidden layers each performed competitively with the best performing model in each case. It is possible that with if more hyper parameter combinations were explored we could identify a neural network that could consistently outperform the Random Forest model on the three datasets.

Table 1.1 : Performance for SMS Spam Detection				
Model	Spam Recall	Spam Precision	Ham Recall	Ham Precision
K-Means Cluster (50)	91.79%	64.22%	94.69%	99.11%
K-Means Cluster (100)	94.07%	63.55%	94.61%	99.38%
Random Forest (50)	**96.53%**	**85.45%**	**97.78%**	**99.52%**
Random Forest (100)	96.22%	84.91%	97.70%	99.48%
Neural Network (5, 5, 0)	79.87%	65.15%	94.74%	97.45%
Neural Network (5, 10, 0)	79.87%	65.15%	94.74%	97.45%
Neural Network (5, 20, 0)	79.87%	65.15%	94.74%	97.45%
Neural Network (10, 5, 0)	87.65%	75.83%	96.33%	98.34%
Neural Network (10, 10, 0)	91.19%	70.49%	95.58%	98.94%
Neural Network (10, 20, 0)	94.35%	77.97%	96.67%	99.27%
Neural Network (20, 5, 0)	91.41%	82.38%	97.31%	98.80%
Neural Network (20, 10, 0)	92.80%	80.91%	97.09%	99.02%
Neural Network (20, 20, 0)	92.83%	82.91%	97.39%	99.01%

Table 1.2 : Performance for Email Spam Detection				
Model	Spam Recall	Spam Precision	Ham Recall	Ham Precision
K-Means Cluster (50)	98.78%	33.68%	88.31%	**99.91**%
K-Means Cluster (100)	95.25%	58.42%	92.30%	99.42%
Random Forest (50)	**99.11**%	**92.72**%	**98.57**%	99.83%
Random Forest (100)	98.88%	92.09%	98.45%	99.79%
Neural Network (5, 5, 0)	44.53%	49.90%	89.76%	87.60%
Neural Network (5, 10, 0)	53.32%	55.09%	90.98%	90.38%
Neural Network (5, 20, 0)	71.20%	56.55%	91.68%	95.44%
Neural Network (10, 5, 0)	64.32%	74.22%	94.70%	91.79%
Neural Network (10, 10, 0)	67.48%	69.02%	93.79%	93.37%
Neural Network (10, 20, 0)	75.97%	77.55%	95.50%	95.11%
Neural Network (20, 5, 0)	70.03%	80.67%	96.02%	93.12%
Neural Network (20, 10, 0)	80.23%	86.07%	97.18%	95.77%
Neural Network (20, 20, 0)	86.25%	84.82%	96.98%	97.31%

Table 1.3 : Performance for Twitter Spam Detection				
Model	Spam Recall	Spam Precision	Ham Recall	Ham Precision
K-Means Cluster (50)	98.87%	35.00%	88.49%	**99.92**%
K-Means Cluster (100)	96.94%	69.60%	94.24%	99.56%
Random Forest (50)	**99.14%**	**92.40%**	**98.50%**	99.84%
Random Forest (100)	**99.14%**	**92.40%**	**98.50%**	99.84%
Neural Network (5, 5, 0)	59.03%	66.00%	93.04%	90.84%
Neural Network (5, 10, 0)	71.54%	71.40%	94.28%	94.32%
Neural Network (5, 20, 0)	79.44%	74.20%	94.91%	96.16%
Neural Network (10, 5, 0)	77.13%	85.00%	96.94%	94.96%
Neural Network (10, 10, 0)	79.85%	83.20%	96.61%	95.80%
Neural Network (10, 20, 0)	84.41%	86.60%	97.31%	96.80%
Neural Network (20, 5, 0)	81.95%	87.20%	97.41%	96.16%
Neural Network (20, 10, 0)	88.54%	89.60%	97.91%	97.68%
Neural Network (20, 20, 0)	91.41%	89.40%	97.89%	98.32%

4.2 Spam Detection On An Unseen Platform

The performance of the models when trained on all datasets except a held out dataset and then evaluated on the held-out dataset is presented in Tables 2.1, 2.2 and 2.3 for SMS Spam, Ling Spam and Twitter Spam marked as held out datasets respectively. Across all three datasets the neural network models had much greater success in identifying spam compared to other models, but they still produced a lot of false positives. Other models skewed heavily in favor of marking messages as ham producing a lot of false negatives. Thus, none of the models tested were capable of

There is another important pattern that emerges from the data. The Spam Precision for all models when evaluated on the SMS Spam dataset (Table 2.1) highlights the novelty of spam from this dataset compared to the other two datasets. Table 2.1 is the only table of the three where Spam Precision reaches single-digit percents. This indicates that spam messages inspired from traditional SMS spam messages on Twitter and in emails might bypass spam detection with high probability. In other words, novel spam on Twitter and in emails would escape detection by the best-performing models for that platform which highlights the weakness of the proposed models in dealing with diversity in spam. This could be an interesting direction for further study.

Table 2.1 : Performance with SMS Spam Held Out				
Model	Spam Recall	Spam Precision	Ham Recall	Ham Precision
K-Means Cluster (50)	**100%**	0%	86.56%	**100%**
K-Means Cluster (100)	**100%**	0%	86.56%	**100%**
Random Forest (50)	37.21%	27.37%	89.17%	92.83%
Random Forest (100)	33.51%	32.84%	89.61%	89.89%
Neural Network (5, 5, 0)	**100%**	0%	86.56%	**100%**
Neural Network (5, 10, 0)	25.00%	00.13%	86.57%	99.94%
Neural Network (5, 20, 0)	37.50%	00.40%	86.60%	99.90%
Neural Network (10, 5, 0)	75.29%	08.54%	87.52%	99.56%
Neural Network (10, 10, 0)	79.37%	06.68%	87.31%	99.73%
Neural Network (10, 20, 0)	51.22%	08.41%	87.42%	98.76%
Neural Network (20, 5, 0)	70.00%	06.54%	87.28%	99.56%
Neural Network (20, 10, 0)	58.52%	10.55%	87.68%	98.84%
Neural Network (20, 20, 0)	56.24%	**31.91%**	**90.10%**	96.15%

Table 2.2 : Performance with Ling Spam Held Out				
Model	Spam Recall	Spam Precision	Ham Recall	Ham Precision
K-Means Cluster (50)	**90.00%**	18.71%	86.00%	**99.59%**
K-Means Cluster (100)	69.27%	51.56%	90.81%	95.44%
Random Forest (50)	68.65%	81.50%	96.17%	92.58%
Random Forest (100)	67.76%	81.70%	96.20%	92.25%
Neural Network (5, 5, 0)	34.45%	91.89%	97.58%	65.13%
Neural Network (5, 10, 0)	25.48%	90.64%	96.19%	47.14%
Neural Network (5, 20, 0)	37.02%	86.28%	96.28%	70.73%
Neural Network (10, 5, 0)	32.36%	94.80%	98.32%	60.49%
Neural Network (10, 10, 0)	34.35%	95.63%	98.65%	63.56%
Neural Network (10, 20, 0)	36.16%	92.31%	97.78%	67.50%
Neural Network (20, 5, 0)	31.09%	96.05%	98.65%	57.55%
Neural Network (20, 10, 0)	33.00%	**97.30%**	**99.12%**	60.61%
Neural Network (20, 20, 0)	45.88%	91.48%	97.88%	78.48%

Table 2.3 : Performance with Twitter Spam Held Out				
Model	Spam Recall	Spam Precision	Ham Recall	Ham Precision
K-Means Cluster (50)	65.77%	29.20%	87.26%	96.96%
K-Means Cluster (100)	**77.03%**	32.20%	87.85%	**98.08%**
Random Forest (50)	62.46%	77.20%	95.21%	90.72%
Random Forest (100)	61.50%	78.60%	95.47%	90.16%
Neural Network (5, 5, 0)	50.94%	75.60%	94.60%	85.44%
Neural Network (5, 10, 0)	38.55%	83.20%	95.63%	73.48%
Neural Network (5, 20, 0)	41.22%	**89.20%**	**97.18%**	74.56%
Neural Network (10, 5, 0)	34.70%	77.60%	94.05%	70.80%
Neural Network (10, 10, 0)	37.81%	81.60%	95.21%	73.16%
Neural Network (10, 20, 0)	42.00%	82.00%	95.55%	77.36%
Neural Network (20, 5, 0)	40.02%	84.60%	96.04%	74.64%
Neural Network (20, 10, 0)	34.81%	84.60%	95.69%	68.32%
Neural Network (20, 20, 0)	36.61%	87.80%	96.61%	69.60%

4.3 Spam Detection Across Multiple Platforms

The performance of the models when trained on all datasets combined evaluated using cross-validation is provided in Table 3.1. The clustering models faced a similar issue as before with very low precision in identifying spam because of a heavy bias towards classifying messages as ham. The best performing model was Random Forest with 100 trees, however, there was no significant difference in the performance of Random Forest model with 50 trees. Larger neural networks were better performing but had significantly lower Spam Precision that the best performing Random Forest model.

Table 3.1 : Performance with All Datasets Merged				
Model	Spam Recall	Spam Precision	Ham Recall	Ham Precision
K-Means Cluster (50)	**100%**	08.73%	86.05%	**100%**
K-Means Cluster (100)	97.23%	14.22%	86.77%	99.93%
Random Forest (50)	96.66%	82.08%	96.90%	99.50%
Random Forest (100)	97.06%	**82.20%**	**96.92%**	99.56%
Neural Network (5, 5, 0)	83.41%	53.47%	92.23%	98.11%
Neural Network (5, 10, 0)	82.09%	57.23%	92.79%	97.78%
Neural Network (5, 20, 0)	80.78%	57.57%	92.83%	97.57%
Neural Network (10, 5, 0)	82.92%	59.77%	93.19%	97.81%
Neural Network (10, 10, 0)	84.69%	60.75%	93.36%	98.05%
Neural Network (10, 20, 0)	85.42%	61.62%	93.50%	98.13%
Neural Network (20, 5, 0)	87.10%	62.83%	93.71%	98.35%
Neural Network (20, 10, 0)	85.78%	65.90%	94.18%	98.06%
Neural Network (20, 20, 0)	85.50%	68.55%	94.60%	97.94%

5 Conclusion

From the three evaluations it was clear that the Random Forest model with 50 trees was the best at dealing with known modes of spam. Random Forest outperformed all other models on all metrics tested on SMS, Email (Ling) and Twitter datasets individually as well as on the mixed dataset. This makes Random Forest classifiers the ideal choice for spam filters targeting Twitter, Email and SMS or any other social media.

However, neural networks performed much better than any other model in identifying spam from novel sources that weren't present in training data. Thus, neural networks are better at identifying new forms of spam than other models. However their performance in this regard is only comparatively better and impractical for use because of the high false positive rate.

There is still scope for more work in the future for this project. The messages were converted to vectors using term frequency but there are other ways of mapping messages to vector spaces used in natural language processing that can be explored. More model hyper parameters can be explored, especially for neural networks. More types of models can also be explored as individual models or in an ensemble as a combination of different models.

References

Bogofilter. (2002). https://bogofilter.sourceforge.io/. (Accessed: 2020-03-01)

Chen, W., Yeo, C. K., Lau, C. T., & Lee, B. S. (2017, Sep). A study on real-time low-quality content detection on twitter from the users' perspective. *Plos One*, *12*(8). doi: 10.1371/journal.pone.0182487

Clement, J. (2019). *Global spam volume as percentage of total e-mail traffic from january 2014 to september 2019, by month.* https://www.statista.com/statistics/420391/spam-email-traffic-share/. (Accessed: 2020-03-01)

Ling spam dataset. (n.d.). http://www.aueb.gr/users/ion/data/lingspam$_public.tar.gz.(Accessed: 2020 - 03$

Mccord, M., & Chuah, M. (2011). Spam detection on twitter using traditional classifiers. *Lecture Notes in Computer Science Autonomic and Trusted Computing*, 175–186. Retrieved from https://link.springer.com/chapter/10.1007/978-3-642-23496-5_13 doi: 10.1007/978-3-642-23496-5_13

Sasaki, M., & Shinnou, H. (2005, Nov). Spam detection using text clustering. *2005 International Conference on Cyberworlds (CW05)*. Retrieved from https://ieeexplore.ieee.org/document/1587549 doi: 10.1109/cw.2005.83

Sms spam collection v. 1. (n.d.). http://www.dt.fee.unicamp.br/ tiago/smsspamcollection/. (Accessed: 2020-03-03)

Study: Twitter bots played disproportionate role spreading misinformation during 2016 election. (2018). https://news.iu.edu/stories/2018/11/iub/releases/20-twitter-bots-election-misinformation.html. (Accessed: 2020-03-01)

Vergelis, M., Shcherbakova, T., & Sidorina, T. (2019). *Spam and phishing in q3 2019.* https://securelist.com/spam-report-q3-2019/95177/. (Accessed: 2020-03-01)