Good word attack on statistical spam filters

Adithya Krishna Murthy
Dept. of Computer Science
Clemson University
Clemson, US
adithyk@g.clemson.edu

Rohan Gangisetty
Dept. of Computer Science
Clemson University
Clemson, US
rgangis@g.clemson.edu

I. Overview of the paper

Introduction

Unsolicited emails are common issue nowadays to each and every individual who uses email services or message services. The emergence of the statistical spam filters gave a deep sigh of relief to the email or message service consumers. But as the statistical spam filters emerged, the spammers also found different ways to circumvent these filters to spam the consumers mailbox with spam messages. One such way is "Good Word attack on statistical spam filters", this technique uses the good words found in the legitimate emails and appends them to the spam email body, thus making them look like a non-spam email or legitimate emails. These attacks are tested against two most famous spam filters: Naive Bayes classification and Maximum Entropy filter (Logistic Regression).

Implementation

This paper deals with two kinds of attacks on spam filters with good words: Active attacks and Passive attacks. Firstly, active attack employs a feedback mechanism while testing a sample email to the spam filter to determine the good words in that email. These words will be later be appended to the spam email to make it look more legit. Secondly, Passive attack is about appending dictionary words, frequent words or frequency ratio words to the spam email. These attacks are tested against two most famous spam filters: Naive Bayes classification and Maximum Entropy filter (Logistic regression).

Passive attacks methodology and result

Choosing words in passive attack was categorized into 3 different ways:

Dictionary attacks: Picking the words from dictionary or randomly and placing them in the spam mails to give it a legitimate look.

Result: Not affective, the email looked more like spam but not ham.

Frequent Word attacks: These are the words which are picked from the legitimate emails and messages, news articles.

Result: Much better than dictionary attacks but requires minimum of 1000 words to be added to spam mail in order to make it look legitimate.

Frequency Ratio attacks: these are the kind of the words, that occur more in the Ham mails compared to the spam mails.

Result: This was huge success; the spam mail was classified legitimate with addition of as minimum as 300 Good words.

Active attacks methodology and result

Choosing the words for passive attacks is done two ways:

First-N words: Choosing the first N words from the list of legitimate words, after adding them, the mail has successfully passed through the spam filters.

Result: Works extremely well against Maxent. First-N word attacks are economical compared to Best-N.

Best-N words: Choosing the most occurring words /most weighted words, after adding to which the mail is successfully past the spam filters.

Result: Best-N attacks works extremely well against Naïve Bayes filter. Best-N attacks are very expensive.

The author also concludes the paper that the MaxEnt gives a better performance compared to the Naïve Bayes classifier because Naive Bayes is called as generative model, trained to increase the likelihood of the training data and Maxent filter is called discriminative model, trained to increase the likelihood of the class labels given the features.

II. Phase 1 : Data collection and analysing data

In the Phase 1, the author utilizes the dataset from spamarchive.org from the month of April 2004. But neither the dataset nor the website mentioned in citation of the paper seems to be valid anymore. Hence, to make the project doable we have chosen an alternate dataset. We have chosen the dataset which contains spam messages/ SMS's instead of Spam emails. As part of the preparation of the dataset for the classification, we have arranged the dataset in such a way that the first column of the data contains whether its spam / ham (legitimate). This dataset is in tab separated values, this can be read by read_csv using the separator "\t". The dataset consists of 5572 records and this is split into training and testing data (4,179 SMS/text messages as training data and 1,396 SMS/text messages as testing data). The dataset is further is split into training and testing dataset such that 1/3 of the data was considered for testing and 2/3 for training.

III. Phase two: Training and testing the dataset

In the Phase 2, the data obtained from Kaggle [1] is split up appropriately and classified using Naïve Bayes, Decision tree and Logistic Regression (MaxEnt) Classifiers. The Figure 1 and Table 1. Shows the result of the classification.

```
(base) Adithyas-MBP:ML_Latest akm$ python3.7 MLIndex.py
Training Dataset: 4179
Testing Dataset:  1393
NB Accuracy: 98.56424982053123
DTree Accuracy with maximum entropy: 96.4824120603015
Logistic Regression AKA MaxEnt: 98.56424982053123
(base) Adithyas-MBP:ML_Latest akm$
```

Figure: 1

It can be clearly inferred that for the given dataset, the accuracy (denoted a percent for Ham emails) for the Naïve Bayes and MaxEnt gives accuracy of 98.56%, whereas the Decision Tree gives better response with accuracy of 96.48%.

	Actual	Naïve Bayes	Decision Tree	MaxEnt
Ham	89.1%	98.5%	96.48%	98.5%
Spam	10.9%	1.5%	3.52%	1.5%

Table: 1

IV. Passive Attacks

Dictionary Words

The Dictionary Dataset are not readily available in corpus to be downloaded and tested; the dictionary dataset has to be built on top of the existing dataset by adding the good words (positive words) from dictionary embedded the spam messages. We built the dictionary dataset by downloading the personal spam email subjects from Gmail and infused it with the dictionary words which were downloaded from internet [2]. We came up with a set of 110 good word spam emails (dictionary dataset). We then appended these to the training data set for testing the behavior of the classifier on these dictionary infused spam messages. The Results obtained are shown in Figure 2.

```
(base) Adithyas-MBP:ML_Latest akm$ python3.7 MLIndex.py
Training Dataset: 4179
Testing Dataset:  1500
NB Accuracy: 96.26666666666667
DTree Accuracy with maximum entropy: 94.66666666666667
Logistic Regression AKA MaxEnt: 96.46666666666667
(base) Adithyas-MBP:ML_Latest akm$
```

Figure 2

When we examined the results carefully, the classifiers Naïve Bayes, Logistic regression and Decision Tree classifies the dataset more spammy. This result is approximately the same as obtained by the author in the paper. There is a certain increase in spam count after adding dictionary words.

	Naïve Bayes	Decision Tree	MaxEnt
Ham	96.2%	94.6%	96.4%
Spam	3.8%	5.4%	3.6%

Table: 2

This Table 2 infers that, for dictionary attack, the expected number of words is negative, indicating that random dictionary words makes an email look more like spam, not less. As Graham-Cumming (2004) points out, spam is now the majority of email and many spams have random words in them, so a random word is actually more likely to be spammy than good.

Frequency Ratio words

These are the kind of words that appear in our Dataset only in the Ham mails but not in the spam mails. We made a comprehensive list of the top 100 words Frequency ratio words (Figure 3) and we downloaded the personal spam emails subject and infused these frequency ratio words to the dataset.

Figure: 3

We first appended all the frequency ratio words (dataset) to the testing dataset and checked the performance of the Classifiers. Figure 4 and Table 3 shows the performance of the classifiers.

```
(base) Adithyas-MBP:ML_Latest akm$ python3.7 MLIndex.py
Training Dataset: 4224
Testing Dataset:  1455
NB Accuracy: 96.56357388316151
DTree Accuracy with maximum entropy: 95.05154639175257
Logistic Regression AKA MaxEnt: 96.63230240549828
(base) Adithyas-MBP:ML_Latest akm$
```

Figure:4

	Naïve Bayes	Decision Tree	MaxEnt
Ham	96.5%	95.0%	96.6%
Spam	3.5%	5.0%	3.4%

Table:3

This directly indicates that the classifiers were not able to detect some of the spam messages. Hence, classified them as Ham but not Spam. Now, we split-up the frequency ratio dataset into training data and testing data and added to their subsets respectively. Now, both the training and testing dataset contains frequency ratio words. We then checked the performance of the classifier; classifier now has been trained

on the dataset which contains the frequency words and tested the classifier against the testing data. The classifier yielded better responses than earlier. Results as depicted (Figure 5 and Table 4). We obtained approximately the same result as mentioned by the author in the paper.

```
(base) Adithyas-MBP:ML_Latest akm$ python3.7 MLIndex.py
Training Dataset: 4224
Testing Dataset:  1562
NB Accuracy: 94.81434058898847
DTree Accuracy with maximum entropy: 93.40588988476313
Logistic Regression AKA MaxEnt: 94.75032010243278
(base) Adithyas-MBP:ML_Latest akm$
```

Figure:5

	Naïve Bayes	Decision Tree	MaxEnt
Ham (before)	96.5%	95.0%	96.6%
Spam(before)	3.5%	5.0%	3.4%
Ham (after)	94.81%	93.4%	94.75%
Spam (after)	5.19%	6.6%	5.25%

Table:4

FREQUENT WORDS

As Dictionary words, the frequent words are also not readily available in corpus to be downloaded and tested. The Frequent words has to be built from dataset by extracting most frequently occurring words from the dataset. We built the frequent words dataset by downloading the personal spam email subjects from Gmail and infused it with the frequent words. We came up with a set of 107 good word spam emails (frequent word dataset). We then appended these to the training and testing data set for testing the behavior of the classifier. We came up with the results as shown in Figure 6

```
(base) Adithyas-MBP:ML_Latest akm$ python3.7 MLIndex.py
Training Dataset: 4179
Testing Dataset:  1500
NB Accuracy: 96.66666666666667
DTree Accuracy with maximum entropy: 95.33333333333334
Logistic Regression AKA MaxEnt: 96.73333333333333
(base) Adithyas-MBP:ML_Latest akm$
```

Figure:6

This yielded better performance than Dictionary words but not as good Frequency ratio words.

V. CONCLUSION

Upon examining the above test results, we can conclude that the Good word attack works well with frequency ratio words. But these classifiers perform better with more retraining. We can also conclude that among all the classifiers Decision tree yields better performance and then Logistic regression and at last Naïve Bayes.

VI . REFERENCES

[1] Spam messages Dataset
https://www.kaggle.com/uciml/sms-spam-collection-dataset
[2] Positive word list
https://positivewordsresearch.com/list-of-positive-words
[3] Code in GitHub:
https://github.com/Adithya12121992/SMSSpamClassification