DECEMBER 17, 2018

NO MORE SPAM

EMAIL CLASSIFICATION EXPERIMENTS

BRANDON CROARKIN
MICHELLE MAK
TENG SONG (T.S.) YEAP

Table of Contents

OVERVIEW **2**

DATA PRE-PROCESSING **2**

Creating Functions 2
Examine the Texts 2

EXPERIMENTATION **3**

Classifying with Unigram Features **3**
Unigram with Non-Alpha Characters Removed 4
Unigram with Negation Words Removed 4
Unigram with Stop Words Removed 4
Unigram with Negation and Stop Words Removed 5
Classifying with Bigram Features **5**
Bigram with Non-Alpha Characters Removed 5
Bigram with Stop Words Removed 5
Bigram with Negation Words Removed 5
Classifying with POS Tag Features **5**
POS Tag with Non-Alpha Characters Removed 5
POS Tag with Stop Words Removed 5
POS Tag with Negation Words Removed 5
Sentiment Analysis **6**
Sentiment Analysis with Non-Alpha Characters Removed **6**
Sci-Kit Learn with Unigram Features **6**
Term Frequency Inverse Data Frequency (TFIDF) **6**

CONCLUSION **6**

Overview

Email is an integral part of our daily lives, for both business and personal reasons. It is the quickest, most direct form of communication, as well as one of the cheapest. Therefore, it is not uncommon to find inboxes inundated with irrelevant promotions, subscriptions, and spam.

This investigation seeks to determine what the best method is for classifying emails as spam or non-spam (labeled "ham" in the experiments). In addition, it will determine a list of top words that help differentiate between the two types of texts. The dataset is taken from the Enron public email corpus, with 3,672 regular emails and 1,500 spam emails.

Data Pre-Processing

Creating Functions

Functions are created in to form a baseline for each method of experimentation. The methods used in this investigation include testing: unigram features, bigram features, part of speech (POS) tags, sentiment analysis, and term frequency inverse data frequency documentation. In addition, filters for non-alphabetical characters, negation words (as in to negate the words following a negation words), and standard stop words. These will be used throughout the investigation to find the combination that produces the best accuracy.

Next, a function is created to train, test, and find the mean accuracy of each method for cross validation. This function takes the number of folds, the feature sets it iterates over the folds using different sections for training and testing, and prints the accuracy for each fold.

The final function that prepares the texts for examination is creates a word list out of the spam and ham datasets. The function creates two lists of 1,500 randomized words of each type of text to be tested.

Examine the Texts

Begin examining the texts by gathering all the words from the emails and putting them into frequency distributions. Each of the following corpuses are comprised of the top 2,000 most frequently used words according to the feature parameters and are saved separately so that they are ready for different levels of experimentation.

The initial observation produces a corpus of all words without the use of any filters. Here are the top 50:

Figure 1

```
['-', '.', ',', '/', ':', 'the', 'to', 'and', 'of', 'a', 'for', 'ect', '?', 'in', 'you', '@', 'this', 'is', 'on',
'=', 'i', "'", ')', '(', 'Subject', 'be', '!', 'your', 'hou', 'that', 'enron', 'with', 'we', 'from', 'have', '$', 'wi
ll', 's', 'are', 'or', 'as', 'at', ';', 'it', '3', 'not', '2000', 'com', '_', 'if']
```

Next, apply the filters one by one to examine their effects on the corpus. The first filter to be applied is stop words – ensure that the standard stop word filter and negation filters do not overlap. The filter created for this is "newstopwords." Here are the top 30 words with stop words removed:

Figure 2

```
['-', '.', ',', '/', ':', 'ect', '?', '@', '=', "'", ')', '(', 'Subject', '!', 'hou', 'enron', '$', ';', '3', 'not',
'2000', 'com', '_', 'please', '``', '1', '2', '00', '%', '|']
```

Negation word filter is applied next. However, it looks like the negation word filter does not reveal any immediate difference from the stop word corpus. Here are the top 30 words with this parameter:

Figure 3

```
['-', '.', ',', '/', ':', 'ect', '?', '@', '=', "'", ')', '(', 'Subject', '!', 'hou', 'enron', '$', ';', '3', 'not',
'2000', 'com', '_', 'please', '``', '1', '2', '00', '%', '|']
```

Lastly, the non-alphabetic characters filter is applied. With punctuations and symbols removed, a larger variety of words now appear in the corpus. Here are the top 100 words from the non-alphabetic corpus:

Figure 4

```
['Subject', 'all', 'me', 'ds', 'here', 'paliourg', 'user', 'id', 'compendia', 'date', 'tue', 'oct', 'mime', 'versio
n', 'content', 'type', 'multipart', 'alternative', 'boundary', 'content', 'type', 'text', 'plain', 'content', 'transf
er', 'encoding', 'bit', 'why', 'pay', 'more', 'when', 'you', 'can', 'enjoy', 'the', 'best', 'and', 'cheapest', 'pill
s', 'online', 'nearly', 'types', 'to', 'choose', 'which', 'makes', 'ours', 'pharmacy', 'the', 'largest', 'and', 'th
e', 'best', 'available', 'no', 'appointments', 'no', 'waiting', 'rooms', 'no', 'prior', 'prescription', 'required',
'see', 'why', 'our', 'customers', 're', 'order', 'more', 'than', 'any', 'competitor', 'this', 'is', 'time', 'mai', 'i
ng', 'no', 'removal', 'are', 're', 'qui', 'red', 'Subject', 'you', 'want', 'a', 'quicker', 'computer', 'spyware', 'st
ays', 'resident', 'in', 'memory', 'just', 'use', 'this', 'to', 'remove']
```

Now that the corpuses are prepared, experimentation can begin.

Experimentation

Classifying with Unigram Features

The first experiment examines the word sets using unigram features. The baseline is established by using the Naïve Bayes classifier on the original feature set, without any filters. The accuracy for this level of classification is already at approximately 95.3%. Here are the top 30 most informative features:

Figure 5

```
Most Informative Features
                    V_cc = True              ham : spam   =     32.6 : 1.0
                  V_2001 = True              ham : spam   =     24.5 : 1.0
              V_attached = True              ham : spam   =     24.5 : 1.0
                   V_ect = True              ham : spam   =     22.3 : 1.0
                   V_hou = True              ham : spam   =     21.6 : 1.0
                  V_deal = True              ham : spam   =     17.3 : 1.0
                    V_28 = True              ham : spam   =     16.5 : 1.0
                  V_file = True              ham : spam   =     15.7 : 1.0
                 V_money = True             spam : ham    =     15.5 : 1.0
                V_robert = True              ham : spam   =     12.1 : 1.0
                 V_offer = True             spam : ham    =     11.8 : 1.0
                    V_04 = True              ham : spam   =     11.3 : 1.0
                V_online = True             spam : ham    =     10.6 : 1.0
                    V_02 = True              ham : spam   =      9.9 : 1.0
                     V_* = True             spam : ham    =      9.7 : 1.0
                  V_most = True             spam : ham    =      9.4 : 1.0
                   V_gas = True              ham : spam   =      9.4 : 1.0
                V_friday = True              ham : spam   =      9.1 : 1.0
                  V_corp = True              ham : spam   =      8.9 : 1.0
                  V_2000 = True              ham : spam   =      8.9 : 1.0
                   V_low = True             spam : ham    =      8.8 : 1.0
       V_international = True             spam : ham    =      8.8 : 1.0
                  V_site = True             spam : ham    =      8.8 : 1.0
                     V_u = True             spam : ham    =      8.8 : 1.0
                V_remove = True             spam : ham    =      8.8 : 1.0
             V_questions = True              ham : spam   =      8.6 : 1.0
                     V_j = True              ham : spam   =      8.6 : 1.0
                     V_. = False            spam : ham    =      8.5 : 1.0
               V_special = True             spam : ham    =      8.2 : 1.0
                 V_works = True             spam : ham    =      8.2 : 1.0
```

It appears that the presences of words such as money, offer, online, *, most, low, international, site, u, remove, special, and works are the best indications that an email may be spam. Additionally, the lack of periods is another giveaway.

Another interesting observation is that while most words in non-spam email seem business related, the letter "j" appears to be another common feature of ham emails. This is possibly because the letter "j" signifies a smiley face in Microsoft Outlook emails and humans are more likely to use emojis in their emails.

Unigram with Non-Alpha Characters Removed

The Unigram feature set it tested once again, but this time without non-alphabetical characters. The number of folds for this round is set to 5, and the average accuracy for this classification drops to around 91.07%. By dropping the punctuation and numbers, a few more words, like such, looking, and internet show up as informative features that indication spam emails. However, based on the accuracy results, these three words are less useful than the *, 2000, and lack of periods.

Figure 6

```
Most Informative Features
                    V_cc = True              ham : spam   =     32.6 : 1.0
             V_attached = True              ham : spam   =     24.5 : 1.0
                  V_ect = True              ham : spam   =     22.3 : 1.0
                  V_hou = True              ham : spam   =     21.6 : 1.0
                 V_deal = True              ham : spam   =     17.3 : 1.0
                 V_file = True              ham : spam   =     15.7 : 1.0
                V_money = True             spam : ham    =     15.5 : 1.0
               V_robert = True              ham : spam   =     12.1 : 1.0
                V_offer = True             spam : ham    =     11.8 : 1.0
               V_online = True             spam : ham    =     10.6 : 1.0
                 V_most = True             spam : ham    =      9.4 : 1.0
                  V_gas = True              ham : spam   =      9.4 : 1.0
               V_friday = True              ham : spam   =      9.1 : 1.0
                 V_corp = True              ham : spam   =      8.9 : 1.0
                  V_low = True             spam : ham    =      8.8 : 1.0
        V_international = True             spam : ham    =      8.8 : 1.0
               V_site = True               spam : ham    =      8.8 : 1.0
                  V_u = True               spam : ham    =      8.8 : 1.0
             V_remove = True               spam : ham    =      8.8 : 1.0
          V_questions = True               ham : spam   =      8.6 : 1.0
                  V_j = True               ham : spam   =      8.6 : 1.0
            V_special = True               spam : ham    =      8.2 : 1.0
              V_works = True               spam : ham    =      8.2 : 1.0
               V_save = True               spam : ham    =      8.2 : 1.0
                 V_am = True               ham : spam   =      7.8 : 1.0
            V_changes = True               ham : spam   =      7.7 : 1.0
           V_thursday = True               ham : spam   =      7.7 : 1.0
               V_such = True               spam : ham    =      7.6 : 1.0
            V_looking = True               spam : ham    =      7.6 : 1.0
           V_internet = True               spam : ham    =      7.6 : 1.0
```

Unigram with Negation Words Removed

The negation word filter produces an average accuracy of about 95.07%, which is only slightly lower than the baseline. It looks like the re-introduction of non-alphabetic characters helps to raise accuracy, but negation words also play an important part in classification.

Unigram with Stop Words Removed

Using the stop words filter resulted in an average accuracy of 94.03%, which is still lower than the baseline accuracy that did not sure any filters. The trend seems to show that the more the number of features are removed, the less accurate the classifier gets.

Unigram with Negation and Stop Words Removed

The last attempt to surpass the baseline accuracy is to combine the negation and stop words filters. However, the average accuracy for this experiment is the same as the negation words – 95.07%. This defeats the previous theory about removing more and getting less accurate results. It seems that the combination of removing negation words and stop words filters together helps to balance out the lower accuracy of just using the stop word filter.

Classifying with Bigram Features

Naturally, investigation of bigram features will follow the unigram feature experiment. We use the NLTK collocations bigram as the bigram finder. The baseline established for bigram features without filters is 95.55%. This is slightly higher than the accuracy of the unigram baseline.

Bigram with Non-Alpha Characters Removed

Once again, we attempt to experiment by removing non-alphabetical characters. This time, the accuracy drops slightly to 95.4%. So far, the results for bigram features remain consistent to that of the unigram in that the baseline without filters seems to provide the best results.

Bigram with Stop Words Removed

Removing stop words renders results of 95.96% accuracy! This is by far the best accuracy we have seen so far and is very surprising, since when testing unigrams, removing stop words produced the worst results.

Bigram with Negation Words Removed

When the negation words are removed for bigram features, the mean accuracy further increases to 96.47%, which sets the new record. In both instances of the unigram and bigram, the negation word filter seems to provide better results than using stop words alone.

Classifying with POS Tag Features

Part of speech tagging is the next experiment attempted in the investigation. The function will look out how many parts of speeches are in the test and training data set and attempt to classify the emails. For example, the baseline experiment establishes that there are 32 nouns, 11 verbs, 17 adjectives, and 4 adverbs in the first sentence in the data set. The baseline accuracy is a mean of 96.3%.

POS Tag with Non-Alpha Characters Removed

The accuracy when the non-alphabetic filter is applied is 96.26%. This result is very close to the baseline.

POS Tag with Stop Words Removed

Interestingly enough, removing stop words renders the same results as the non-alphabetic character filter – 96.26%. However, upon examining the accuracy of each fold, it is clear that each round had a different accuracy from the non-alphabetic experiment.

POS Tag with Negation Words Removed

Furthermore, POS tagging with negation words removed also produces the exact same accuracy score – 96.26%. But when examining the accuracy of each of the folds, the scores are actually identical to that of the stop word POS tag experiment.

Sentiment Analysis

The ability of sentiment to classify emails is the next experiment. To do this, each word feature needs to be assigned a sentiment score that determine how positive or negative a word is. To perform the sentiment analysis, a subjectivity file that contains the sentiment scores for words was read in. With these sentiment scores, a new feature set is created that labels a word with a sentiment score as opposed to a boolean tag. With these sentiment features added the accuracy reached is 95.3%.

Sentiment Analysis with Non-Alpha Characters Removed

The accuracy when the non-alphabetic filter is applied to the sentiment analysis is also 95.3%. This result is the same as the baseline. The lack of a change is not surprising since these non-alpha characters do not match to a word in the sentiment file and therefore should not affect the classification output.

Sci-Kit Learn with Unigram Features

To perform the unigram feature analysis using the Sci-Kit Learn package in Python, first the data has to be put in the proper format. First, the feature sets defined in NLTK are written to a csv with forbidden characters like ",", " ' ", and " " " removed. Each feature set is converted to a line in the file with comma separated feature values. Each feature value is converted to a string. For booleans this is the words true and false and for numbers, this is the string with the number. The result is a data frame with a row representing one email and columns representing whether the feature is contained in the email and the last column representing the label as "ham" or "spam".

Figure 7

	V_-	V_.	V_/	V_CM	V_:	V_the	V_to	V_ect	V_and	V_of	...	V_teams	V_mean	V_memo	V_liquids	V_situation
0	False	True	True	True	True	True	True	False	False	True	...	False	False	False	False	False
1	True	True	True	True	True	True	True	False	True	False	...	False	False	False	False	False
2	True	True	True	True	True	True	True	False	True	True	...	False	False	False	False	False
3	True	True	False	True	True	True	False	False	True	False	...	False	False	False	False	False
4	True	True	False	True	True	True	True	False	True	True	...	False	False	False	False	False

5 rows × 2001 columns

After the features are in the proper format, the data is split into a training and test set and a Random Forest model was used to predict the label of the email using the features. This process resulting in an accuracy of 97.6%.

Term Frequency Inverse Data Frequency (TFIDF)

A term frequency experiment is created using a function that produces the normalized frequency score of the words in the data set. This method will count the number of times a word appears throughout the dataset and classify the emails based on the normalized frequency score of each word. The baseline accuracy of this experiment is 95.33%.

Conclusion

Right off the bat, the baseline accuracy of all methods without any filters provides satisfactory results. The highest accuracy emerged out of the Sci-Kit Learn experiment, at 98%. This is likely because the Sci-Kit Learn experiment used a different algorithm, Random Forest, to classify. Random Forest tends to be a stronger classification algorithm than Naïve Bayes and this likely

contributes to the better performance compared to the NLTK classification algorithms that only use Naïve Bayes.

It was also fascinating to see the large difference in results between unigram and bigram methods, especially with the different filters applied. When it comes to unigram, it seems that more to work with is better because every detail from letters to characters contributes to the classification. Whereas with bigram, since it is looking at word pairs, the negation word filter actually produced the best result.

Furthermore, part of speech tagging resulted in the exact same score across all the filtered versions. This method was both accurate, averaging above 96%, and consistent.

Overall, it is clear that deciphering between spam and non-spam emails is quite achievable feature – no matter the method.