Darrell Nelson

05/28/2019

IST 664

NLP Final Project Report

Overview

A Naïve-Bayes classification model was used to determine what types of feature sets best classify text data. The dataset chosen is the email spam dataset from the Enron public email corpus. This corpus contains 1,500 spam emails and 3,672 non-spam (labelled as "ham") emails. The task is to run a Naïve-Bayes (NB) classifier on different feature sets with an 80/20 split for training and testing respectively. Then, compare the different methodologies.

Processing

The Enron public email corpus was downloaded with folders created for each category: spam and non-spam. The function "processspamham" reads the spam and ham files and creates a list consisting of the contents of the emails with their appropriate labels. The words in the email document list were then extracted into their own list. The most common 1,500 words in the entire list were chosen and paired with their word frequency count to make a "word_features" attribute. This attribute is then fed into a document features definition which formats the dataset to allow for different types of processing. The document features chosen were: 1) the baseline which uses just the word frequency counts 2) Bigrams where the top 500 occurring bigrams are looked at by counts and 3) removing stopwords from the baseline document feature. These features were then formatted into feature sets and tested via 10-fold cross validation with a NB classifier. The same model is then used to output the precision, recall, and F-measure scores.

Features

Baseline

This feature is based off the principle that spam emails will have different vocabulary than ham/non-spam emails. It uses the word frequency distribution to figure out how many words in the set are in each email and with what frequency. With each row representing an email, the data is then given to the featuresets function which adds the appropriate spam/non-spam labels. This labelled data is then randomly separated into training and testing data at an 80/20 split respectively. The data is trained on the NB classifier and tested using a 10-fold cross validation.

Bigram Filter

This feature is based off the principle that spam emails will use different series of words than ham/non-spam emails. It takes the tokenized word list generated for each email and generates the top 1000 bigrams (a pair of consecutive words) and their frequency distribution. With each row representing an email, the data is then given to the featuresets function which adds the appropriate spam/non-spam

labels. This labelled data is then randomly separated into training and testing data at an 80/20 split respectively. The data is trained on the NB classifier and tested using a 10-fold cross validation.

Stopwords Filter

This feature is based off the principle that spam emails will have different vocabulary than ham/non-spam emails. But, unlike the baseline feature, this feature removes filler words. Words such as: the, at, a, etc. These words will no doubt be ranked the highest among both the spam and ham emails. So, getting rid of them should provide a clearer contrast between the vernacular between the two sets. It uses the word frequency distribution to figure out how many words in the set are in each email and with what frequency. It then removes the English stopwords from this dataset. With each row representing an email, the data is then given to the featuresets function which adds the appropriate spam/non-spam labels to each row. This labelled data is then randomly separated into training and testing data at an 80/20 split respectively. The data is trained on the NB classifier and tested using a 10-fold cross validation.

Classification Experiments

The outputs from all three different feature sets are labeled below. Each output is a screenshot of the Anaconda Prompt console and gives the fold size for each k-fold cross validation, the accuracy of each k-fold validation, the total mean accuracy, confusion matrix, precision, recall, and F1 measures. The mean accuracy for each model was ~94%, which is good based on the differences between the feature sets provided. This could be attributed to the algorithm finding a few distinct features that are only present in spam or non-spam emails. Another explanation could be that the number of non-spam emails (3,672) greatly outweighs the number of spam emails (1,500) which is skewing the accuracy output.

To combat this, the precision, recall, and F1 measures were also calculated. Precision in this context is the proportion of true positive identifications that were correct for a class label. Precision answers the question: Out of the predicted labels for this class, what percentage of them were actually correct? The feature recall, in this context, is the proportion of positives for a given class that were identified correctly. Recall answers the question: Out of the correct labels for a class, how many of them were properly predicted?

recall = TP / (TP + FP) *(the percentage of actual yes answers that are right)*
precision = TP / (TP + FN) *(the percentage of predicted yes answers that are right)*

These two measures were also combined using a harmonic mean called the F-measure.

F-measure = 2 * (recall * precision) / (recall + precision)

Baseline	Precision	Recalll	F1
spam	0.997	0.836	0.909
ham	0.92	0.999	0.957
Bigram			
spam	0.999	0.808	0.894
ham	0.905	1	0.95
Stopwords			
spam	0.998	0.828	0.905
ham	0.914	0.999	0.955

The precision metric for all 3 feature sets was over 90% accurate in predicting whether an email was spam or non-spam. All 3 feature sets had a 99.7% or higher chance of properly labeling the emails. This raises the question, Did the spam emails contain non-sensical words and phrases? Looking at the example (Email: 0017.2003-12-18.GP.spam) below it appears that spam emails tend to have misspellings and grammatical issues like not having a space between “now” and “the” in the beginning of the sentence.

Subject: get that new car 8434
people nowthe weather or climate in any particular environment can change and affect what people eat and how much of it they are able to eat .

Email: 0017.2003-12-18.GP.spam

With the recall metric, there appears to be a disparity. All 3 feature sets were able to label >99% of their non-spam predictions correctly, however the spam emails were a lot tougher to navigate with all 3 feature sets labeling < 84% correctly.

Although the overall accuracy of the 3 models is ~94%, the recall, and precision metric allow for a more robust comparison between the three. The best model based on the harmonic mean between recall and precision is the baseline. The baseline feature set did the best job at predicting a higher number of spam emails which is why its F1 measure is slightly higher.

Console Output: Baseline Feature Set

```
Each fold size:  517
0 0.9477756286266924
1 0.9264990328820116
2 0.9381044487427466
3 0.9381044487427466
4 0.941972920696325
5 0.9381044487427466
6 0.9477756286266924
7 0.9439071566731141
8 0.9284332688588007
9 0.9535783365570599
Cross validation mean accuracy:  0.9404255319148935
     |              s |
     |      h       p |
     |      a       a |
     |      m       m |
-----+----------------+
 ham | <65.2%>  5.7% |
spam |   0.1% <29.0%>|
-----+----------------+
(row = reference; col = test)

        Precision       Recall          F1
spam          0.997      0.836      0.909
ham           0.920      0.999      0.957
```

Console Output: Bigram Feature Set

```
Each fold size:  517
0 0.9497098646034816
1 0.9361702127659575
2 0.9264990328820116
3 0.9613152804642167
4 0.9245647969052224
5 0.9381044487427466
6 0.9381044487427466
7 0.9535783365570599
8 0.9323017408123792
9 0.9593810444874274
Cross validation mean accuracy:  0.941972920696325
     |              s |
     |      h       p |
     |      a       a |
     |      m       m |
-----+----------------+
 ham | <64.6%>  6.8%  |
spam |   0.0% <28.6%> |
-----+----------------+
(row = reference; col = test)

        Precision       Recall           F1
spam          0.999       0.808       0.894
ham           0.905       1.000       0.950
```

Console Output: Removing Stopwords Feature Set

```
Each fold size:  517
0 0.941972920696325
1 0.9381044487427466
2 0.9381044487427466
3 0.941972920696325
4 0.9400386847195358
5 0.9439071566731141
6 0.941972920696325
7 0.9516441005802708
8 0.9497098646034816
9 0.9342359767891683
Cross validation mean accuracy:  0.9421663442940039
     |              s |
     |      h       p |
     |      a       a |
     |      m       m |
-----+----------------+
 ham | <64.7%>  6.1%  |
spam |   0.1% <29.2%> |
-----+----------------+
(row = reference; col = test)

        Precision       Recall          F1
spam          0.998      0.828      0.905
ham           0.914      0.999      0.955
```