CS 547 Assignment 2

Text Categorization for Detecting Email Spam

Tingjun Li

2. Evaluation (30 points)

First, you need to evaluate the result in "testOutput" by using the command "perl TC_eval.pl testOutput 0.5". (5 points)

```
sslab21 127 $ perl TC_eval.pl testOutput 0.5
Total Number of test emails are 986
The value of True Positive is 150
The value of True Negative is 813
The value of False Postive is 2
The value of False Negative is 21
```

Second, please read through the perl script "TC_eval.pl" and explain the meaning of the output result from the previous step (e.g., what does true positive mean and what is 0.5 in the command) (5 points)

True Positive: Correctly identified relevant email
True Negative: Correctly identified irrelevant email
False Positive: Incorrectly identified relevant email
False Negative: Incorrectly identified irrelevant email
0.5: The threshold of probability of relevant document used by determining a document is relevant or not.

Third, calculate two new measures of Precision and Recall (5 points)

Precision = 150 / (150 + 2) = 98.68%
Recall = 150 / (150 + 21) = 87.72%

Fourth, try different thresholds (e.g., 0.99, 0.9, 0.5, 0.1, 0.01, 0.001....); calculate the corresponding precision and recall measures with each threshold; plot a figure to show these values (x-axis, precision, y-axis-recall) (15 points)

0.99

```
sslab21 128 $ perl TC_eval.pl testOutput 0.99
Total Number of test emails are 986
The value of True Positive is 142
The value of True Negative is 814
The value of False Postive is 1
The value of False Negative is 29
```

Precision = 142/(142 + 1) = 99.30%
Recall = 142/(142+29) = 83.04%

0.9

```
sslab21 129 $ perl TC_eval.pl testOutput 0.9
Total Number of test emails are 986
The value of True Positive is 146
The value of True Negative is 814
The value of False Postive is 1
The value of False Negative is 25
Precision = 146/(146 + 1) = 99.32%
Recall = 146/(146+25) = 85.38%
```

0.5

```
sslab21 127 $ perl TC_eval.pl testOutput 0.5
Total Number of test emails are 986
The value of True Positive is 150
The value of True Negative is 813
The value of False Postive is 2
The value of False Negative is 21
Precision = 150/(150 + 2) = 98.68%
Recall = 150/(150+21) = 87.72%
```

0.1

```
sslab21 130 $ perl TC_eval.pl testOutput 0.1
Total Number of test emails are 986
The value of True Positive is 151
The value of True Negative is 813
The value of False Postive is 2
The value of False Negative is 20
Precision = 151/(151 + 2) = 98.69%
Recall = 151/(151+20) = 88.30%
```

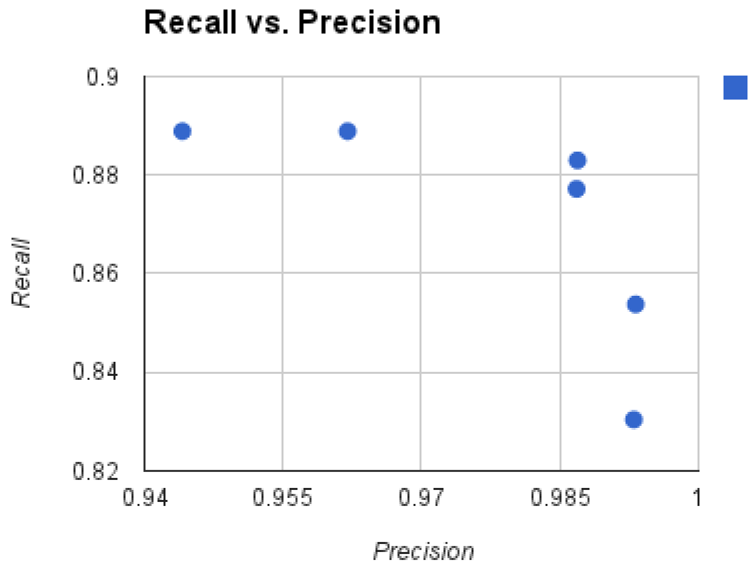
0.01

```
sslab21 131 $ perl TC_eval.pl testOutput 0.01
Total Number of test emails are 986
The value of True Positive is 152
The value of True Negative is 809
The value of False Postive is 6
The value of False Negative is 19
Precision = 152/(152 + 6) = 96.20%
Recall = 152/(152+19) = 88.89%
```

0.001

```
sslab21 132 $ perl TC_eval.pl testOutput 0.001
```

Total Number of test emails are 986
The value of True Positive is 152
The value of True Negative is 806
The value of False Postive is 9
The value of False Negative is 19
Precision = 152/(152 + 9) = 94.41%
Recall = 152/(152+19) = 88.89%



3. Model Analysis (15 points)

The application of TCEval prints out two lists of words with the highest word probabilities in the relevant (i.e., spam) and in the irrelevant (i.e., non-spam) models. Please analyze these words.
First, are the words in the two lists the same or different?

Different

Give some examples of good words for detecting spam and bad words. Can you explain why there are bad words?

Good words such as: univers, paper, subject, informa
Bad words such as: word, book
Reason of there are bad words because that these are common words that people use in conversation, their appearance cannot really help identifying spam.

4. Algorithm Revisit (5 points)

Try your ideas to improve the Naive Bayes algorithm. Some possibilities are: (1) Try different smoothing methods (i.e., MAP, Jelinek Mercer...); (2) Try feature selection (keyword selection); for example, only consider the features (keywords) that are helpful for detecting spam. Full credit (i.e., 5 pts) will be given if you try some interesting algorithm (e.g., feature selection) other than only simple smoothing methods.

Did not implement.