COMP472 - Reports

Thomas Backs[1], Marco Tropiano[2], and Earl Aromin Steven[3]

[1] 27554524 - thomasbacks@gmail.com
[2] 26789331 - tropiano.m@gmail.com
[3] 40004997 - earlaromin@gmail.com

1 Introduction

This project is completed within the COMP 472 course. It aims is to create an unsupervised machine learning program that will take some people feedback on various items they purchased and to classify it in two different group: Positive review or negative review. We used 80% of the data collected to train our machine, and the remaining 20% to test our machine. The analysis of each task of the project are shown in different section.

2 Task 1 Analysis

For the task 1, we created a function called `def train_nb(documents, labels)`, it takes two parament which are the `documents` and the `labels` to train our machine. We also import the `Counter` from the `collections` modules as well to help us to count the number of occurence of each word. We separate the occurence of word in negative review from the occurence in positive review by using the labels already there to teach

```
def train_nb(documents, labels):
neg_word_count = Counter()
pos_word_count = Counter()
neg_total_word = 0
pos_total_word = 0
# we now create our classification
classifier = list(zip(labels, documents))
for c in classifier:
    if c[0] == 'neg':
        neg_word_count.update(c[1])
    else:
        pos_word_count.update(c[1])

neg_total_word = sum(neg_word_count.values())
pos_total_word = sum(pos_word_count.values())
total_word = sum(neg_total_word, pos_total_word)
return neg_total_word, pos_total_word,
    total_word, neg_word_count, pos_word_count
```

3 Task 2 Analysis

The task 2 is seperate in two part, where the first part is to get the score and the second part compares the scores we got and return the appropriate classification (negative or positive review).

The function called `score_doc_label(document, smoothing=0.5)`.
The smoothing paramater is set to default which is 0.5 and can be modified in the function call to suit future need. We use this formula to find our probabilty:

$$score(\text{Negative}) = \log_{10}(P(\text{Negative})) + \sum_{i=1}(P(w_i|\text{Negative}))$$
$$score(\text{Positive}) = \log_{10}(P(\text{Positive})) + \sum_{i=1}(P(w_i|\text{Positive}))$$

Then we take the scores from our formula calcuation above, and we return the exponential of positive and negative scores to make it easier to read:

$$e^{\text{score(Negative)}} \text{ and } e^{\text{score(Positive)}}$$

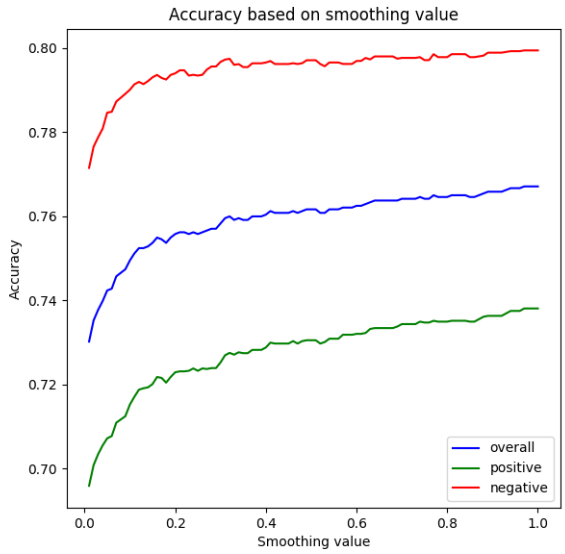
Now this takes us to the part of of the task 2, to perform our probabilities calculation, we call the function called `classify_nb(document, smoothing=0.5)`, this function will call the function previously mentionned in the first part, and will get the return values of the scores for negative and positive probability. It will compares both scores, pick the highest one and will return the fitting label for our review.

4 Accuracy Analysis

We use the 0.5 smoothing for our training data and we end up with this result shown below:

```
Training set accuracy (0.5) :
   Overall accuracy : 0.8712621970412339
   Pos accuracy : 0.8429329291398256
   Neg accuracy : 0.906418998354103
```

The data above show the accuracy of our training data with a smoothing of 0.5. Below there is more datas and graphs based on the evaluation (test) data we have. The first graph show are the overall accuracy based on different smoothing values between 0.1 and 1.
`max overall acc at smoothing value 0.97: 0.767`



5 Issues

Issues encountered...

6 What would you expand

Waht would we expand...