CS 458/535 — Natural Language Processing

Fake News Detection

The dissemination of Fake news always beat out the truth with significant growth. Fake news and false rumors are spreading further and faster, reaching more people, and penetrating deeper into social networks. The objective of this assignment is to address the problem of detecting deceiving information in Urdu language from digital media text.

Your goal in this assignment is to perform Fake News Detection: classifying news articles as fake or real. Recall from the lecture on sentiment analysis that it can be used to extract people's opinions about all sorts of things (tweets, speeches, reviews, blogs) and at many levels of granularity (the sentence, the paragraph, the entire document). Our goal in this task is to look at a news story and classify it as $fake$ or $real$.

You will be using Naïve Bayes, following the pseudocode given in Algorithm 1 and Algorithm 2 using Laplace (add-1) smoothing. Your classifier will use words as features, add the $log-prob$ scores for each token, and make a binary decision between $fake$ and $real$.

Algorithm 1 TrainMultinomialNB($\mathcal{C}, \mathcal{T}$)

```
procedure NaiveBayesTraining($C, T$)
    $V \leftarrow$ ExtractVocabulary(C)
    $N \leftarrow$ CountTexts(T)
    for each $c \in \mathcal{C}$ do
        $N_c \leftarrow$ CountTextsInClass($\mathcal{T}, c$)
        $N_w \leftarrow$ CountWordsInAllTextsOfClass($\mathcal{T}, c$)
        $prior[c] \leftarrow \frac{N_c}{N}$
        $doc_c \leftarrow$ ConcatenateTextsInClass($\mathcal{T}, c$)
        for each $w_i \in \mathcal{V}$ do
            $N_i \leftarrow$ CountTokensOfWords($doc_c, w_i$)
        for each $w_i \in \mathcal{V}$ do
            $condprob[w_i][c] \leftarrow \frac{N_i+1}{N_w+|V|}$
    return $V, prior, condprob$                                ▷ The NB Model
```

You will also explore:

(1) The effects of stop-word filtering. This means removing common words from your train and test sets. We have provided a stopword list in the file:
`stopwords-ur.txt`

(2) The effects of Boolean Naïve Bayes. This means removing duplicate words in each document (review) before training.

Algorithm 2 ApplyMultinomialNB($\mathcal{C}, \mathcal{V}, prior, condprob, t$)

```
procedure NaiveBayesTest($\mathcal{C}, \mathcal{V}, prior, condprob, t$)
    $\mathcal{W} \leftarrow$ ExtractWordsFromText($\mathcal{V}, t$)
    for each $c \in \mathcal{C}$ do
        $score[c] \leftarrow$ log $prior[c]$
        for each $w_i \in \mathcal{W}$ do
            $score[c]$ += log $condprob[w_i][c]$
    return $argmax_{c \in \mathcal{C}} score[c]$                    ▷ The Predicted Class
```

1 Assignment Tasks

Train a Naïve Bayes classifier on the data set provided.

Your task is to implement the classifier training and testing code and evaluate them using Accuracy, Precision, Recall and F_1 measures. Next, evaluate your model again with the stop words removed. Show how this approach affects average accuracy (for the current given data set).

2 Requirements

You are required to implement:

(1) Naïve Bayes classifier.

(2) Boolean Naïve Bayes classifier.

(3) Experiment with/without using stop words.

(4) Metrics for evaluating performance of your classifier.

You are required to submit a single page report within the python notebook that analyzes your runs with/without stop word filtering, and mention if the Boolean Naive Bayes improves your performance or not.

3 Extra Credit

There are sometimes news that use words with negation. Preprocess the negations in the dataset and see if it improves your classification.