**Report on**

**Text Classification**

**Problem Statement**

Implement various text classification models and use them to classify sentences from 2016 presidential debates according to speaker.

I have implemented models as mentioned below for respective problems.

Problem1.

Naïve Bayes Classifier

Problem2.

Binary Naive Bayes Classifier  
 Naïve Bayes Classifier using Bigrams as feature

Classification accuracy of Naive-Bayes is: 52.25 %

Classification accuracy of Binary NB is: 55.75 %

Classification accuracy of Naive-Bayes using Bi-gram is: 56.75 %

**Data Used**

We are using the sentences from 2016 presidential debates.

**Challenges Faced**

Understanding the training of the model and how to use it for classficiation. Explore different models to understand their functionality. The training set provided is not satisfactory huge to accurately classify the test data. Initially I had implemented a Naïve Bayes model without add one smoothing which resulted in low accuracy for the classification model. After incorporation log probability along with Laplace model the accuracy of naïve Bayes classifier improved significantly.

**Methodology**

I have built 3 models

Naïve Bayes classifier, Binary naïve Bayes classifier and naïve bayes classifier with bigram as feature.

**Naïve Bayes Classifier**

To learn the probability P(fc), we’ll assume a feature is just the existence of a

word in the document’s bag of words, and so we’ll want P(wc), which we compute as the fraction of times the word w appears among all words in all documents of topic c. We ﬁrst concatenate all documents with category c into one big “category c” text. Then we use the frequency of w in this concatenated document to give a maximum likelihood estimate of the probability:.

The pseudocode to build the training model and calculate the probability values is as below.

#Get data from teh training corpus

loglikelihood\_w\_c[w][c] = math.log((count\_w\_c[w][c]+1)/(class\_word\_count[c]+1),10)

**Binary Naïve Bayes Classifier**

Let's say we have a text document with N unique words making up a vocabulary V, |V|=N|V|=N. For a bigram language model with add-one smoothing,

The probability of bigram model can be calculated as below:

P(wi|wi−1)=count(wi−1wi)+1/count(wi−1)+|V|

For add-one smoothed bigram counts, we need to augment the unigram count by

the number of total word types in the vocabulary V:

**Naïve Bayes Classifier with Bigrams as features**

In custom model I have used all the features of Laplace Unigram and Bigram language model along with the tri-gram language model. The motivation is to use the best possible feature of different model to increase the accuracy of the spell checker algorithm. The logic of the custom model is explained below.

The probability of tri- gram model is calculated as below:

P(wi|wi−2wi−1)=count(wi−2wi−1wi)+1/count(wi−2wi−1)+|V2|

if count3 > 0: #Biigram

value += math.log(count3)

value -= math.log(self.bigramDict[(word\_1,word\_2)])

elif count2 > 0: # no trigram, but bigram exists

value += math.log(0.4) + math.log(count2)

value -= math.log(self.unigramDict[word\_2])

else: # no trigram or bigram

value += math.log(0.4) + math.log(self.unigramDict[word\_3]+1)

value -= math.log(self.total + (len(self.unigramDict)))

**Results:**

The results of the accuracy for different models are as follows. Accuracy is calculated by , the number of valid corrections, divided by the number of test sentences.

Classification accuracy of Naive-Bayes is: 52.25 %

Classification accuracy of Binary NB is: 55.75 %

Classification accuracy of Naive-Bayes using Bi-gram is: 56.75 %

**Conclusion:**

The above chart clearly concludes that the Naïve Bayes classifer with add one smoothing provide better accuracy when compared to unsmoothed models. The accuracy of the models for bi-gram NB model is better than Naïve Bayes model. In the custom language model when we in-corporate the features of tri-gram model along with the bi-gram Naïve Bayes models the accuracy further increases than the Naïve Bayes and Binary Naïve Bayes.