Lei (Raymond) Chi

**Text Categorization**

This project implements the Naïve Bayes method for text categorization. The program begins by reading the file and its corresponding category. The **tokenize\_file()** function then processes the articles by removing punctuations using **string.punctuation**, tokenizing with **nltk.tokenize**, eliminating stopwords with **nltk.corpus**, and performing stemming using **PorterStemmer**. After tokenization, the program calculates the prior and likelihood values, which are then used in the Naïve Bayes equation to determine the category with the highest probability.

Two types of smoothing methods were explored in this project: Laplacian smoothing and Jelinek-Mercer (JM) smoothing. JM smoothing performed best with corpus 1 and 3, achieving accuracy above 90%, but reduced the accuracy of corpus 2 to only 80%. As a result, Laplacian smoothing was chosen, with a constant alpha of 0.058 (tuned through multiple testings). The results below were obtained using this method. For corpus 2 and 3, the data was split in a 55/45 ratio.

To run the code:

1. Place **CHI\_naive\_bayes.py** in the **/TC\_provided** directory.
2. Run the code.
3. When prompted, enter the name of the file containing the list of labeled training documents (e.g., **./corpus1\_train.labels**).
4. Enter the name of the file containing the list of unlabeled test documents (e.g., **./corpus1\_test.list**).
5. The program will generate a file named e.g., **predicted\_corpus1\_test.labels** in your directory (the **corpus#** is determined by the first word of the file name you input).
6. To check accuracy, use the command: **perl analyze.pl predicted\_corpus1\_test.labels corpus1\_test.labels**.

Corpus 1 performance:

A screenshot of a computer

Description automatically generated

Corpus 2 performance:

A screenshot of a computer

Description automatically generated

Corpus 3 performance:

A screenshot of a computer

Description automatically generated