CSC 405/605 Fall 2019

**Budget Text Analysis**

Project Review Document

Version 3.0

Prepared By;   
 Naseeb Thapaliya

Akash Meghani  
 Unnati Khivsera  
 Sultan Al Bogami  
 Miguel Gaspar Utrera

**Tasks**

Sultan Al Bogami

1. Reconstructed the data hierarchy for the project:

a. Collected the budget documents from 2019-2008.

b. Compressed them to reduce the size of the files.

c. Converted the PDF files into CSV files.

d. Standardized the naming scheme of the data files.

e. Organized the data into folders.

f. Uploaded the data to the project repository.

2. Preprocessed the data for FY2020:

1. Convert the csv files into data frames.
2. Label the data frames adding “organization” column.
3. Combine the data frames and insert a “year” column.
4. Drop unnecessary columns in the combined data frame.
5. Perform text normalization.
6. Produces a single csv file for the year in question.

3. Started the exploratory data analysis.

Miguel Gaspar Utrera

1. Did some analysis on the budget for Durham city for 2019 and 2020
2. My goal was to see how similar the two budget documents were
3. Using NLTK, genism and TFID I was able to use the most frequent words from each document and see their similarity
4. They are 57% similar, I can conclude the budget documents are reused and that is why they are very similar

Unnati Premchand Khivasara

Tasks: My main tasks for this project were to do sentiment analysis on the budget documents, compare the changes and spread of sentiments over budget documents for multiple years and for different cities and counties.

1. Data Extraction and Statistical Evaluation - The csv files created using a tool, were loaded in dataframes. This data was evaluated to get the hold of data for the budget text of different years and cities and associated sentiments.(3 hours)

2. Distribution Modeling - (4 hours)

https://github.com/UNCG-CSE/Budget\_Text\_Analysis/blob/master/src/EmotionAnalysis/Distribution.ipynb

a. The sentiments of Guilford County(2008) were plotted using Histogram and KDE showed that the distribution is of type Poisson. Also KStest proved that Poisson Distribution fits well(pValue=0.99).

b. The sent-count(count of occurance of each words) values were plotted to see the distribution of Charlotte City Budget Document. The distribution proves to be of type Poisson using KDE and KStest(pvalue=1.0)

3. Hypothesis Testing - The budget documents should have variance in sentiments over the years, or in difference cities/counties based on the varying economies. The Hpothesis results shows that onsecutive years will have more similarity in sentiments and also does not differs to a great extent over a long period. (4 hours)

https://github.com/UNCG-CSE/Budget\_Text\_Analysis/blob/master/src/EmotionAnalysis/HypothesisTesting.ipynb

1. H0 : The sentiments for Charlotte Document 2008 and 2020 are same

H1 : The sentiments for Charlotte Document 2008 and 2020 are not same

Two sample test was performed to prove this Hypothesis with threshold 0.05.

Result : pvalue = 0.28 so I accept a null Hypothesis.

1. H0 : The sentiments for Raleigh Document 2014 and 2015 are same

H1 : The sentiments for Raleigh Document 2014 and 2015 are not same

Two sample test was performed to prove this Hypothesis with threshold 0.05.

Result : pvalue = 0.98 so I accept a null Hypothesis.

4. Observing the order of Sentiments - For all counties/cities budget documents the frequencies of all sentiments and emotions were obtained. And it was observed that for each document Positive sentiment value was highest and Disgust occurred with least number of frequencies. Also the plot for comparison shows that when seen in descending order every city shows much similarity. (3 hours)

https://github.com/UNCG-CSE/Budget\_Text\_Analysis/blob/master/src/EmotionAnalysis/CountingWordsFrequency.ipynb

5. Sections Sentiment Coverage - If the distribution of sentiments is observed over pages of a budget document, it clearly shows that for specific sections the concentration of positive and negative sentiment if much higher. Although positive and negative sentiments are seen all over document text like Funds and Services. While the emotions are not seen in few sections at all.(2 hours)

https://github.com/UNCG-CSE/Budget\_Text\_Analysis/blob/master/src/EmotionAnalysis/Statistics.ipynb

4. Machine Learning -Question Asked: How to create a model that can classify and predict sentiments for future documents?

Initially steps of data-clean, preprocess, assigning affinity scores, features creation and vector tokenization (using TF/IDF) is done. The data is split into train and test data, and supervised algorithms as LogisticRegression, RandomForestClassifier and LinearSVC were used to create a model.

X -> feature vectors formed out of sentences of a Guilford County funds section(2008)

Y -> Classified sentiments

https://github.com/UNCG-CSE/Budget\_Text\_Analysis/blob/master/src/EmotionAnalysis/MachineLearning3.ipynb

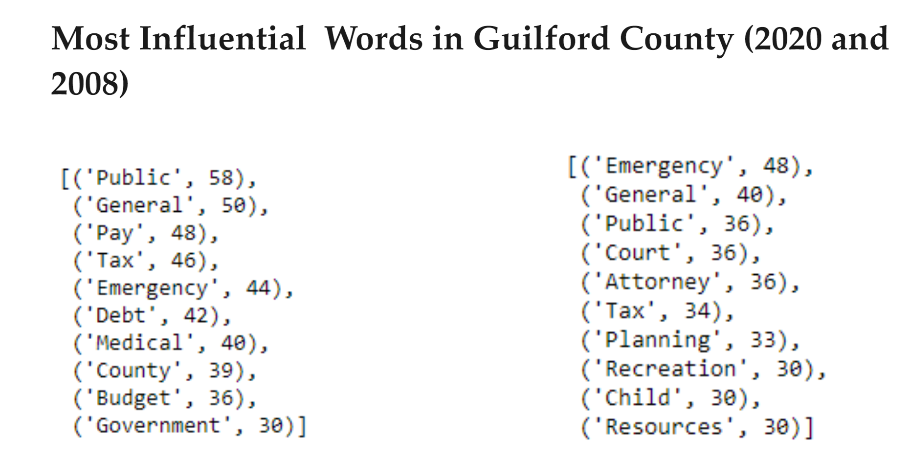
Observation : RandomForestClassifier and LinearSVC classification models proves to predict and classify the sentiments of test data than model using LogisticRegression. These both models predict the results with same accuracy as well as rmse values. While if the proportion of train vs test data is changed the LinearSVC does not show the same accuracy as that of RandomForestClassifier, whose accuracy remain highest and Root mean square error value is lowest.(8 hours)

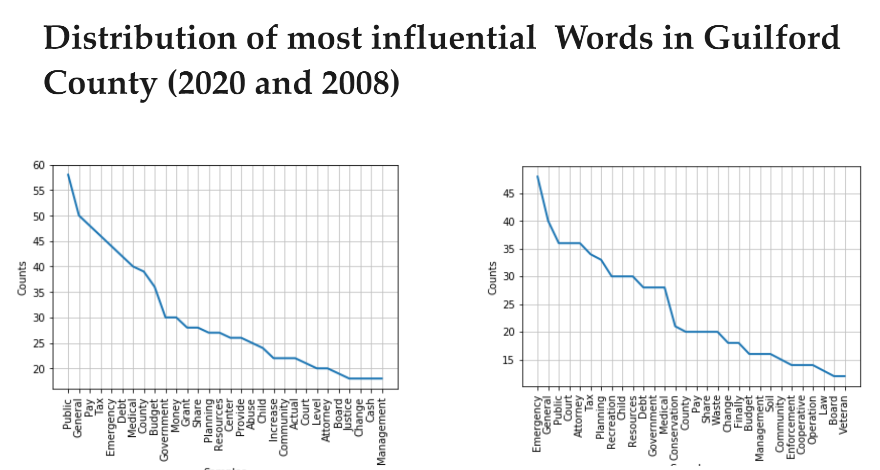
Akash Meghani

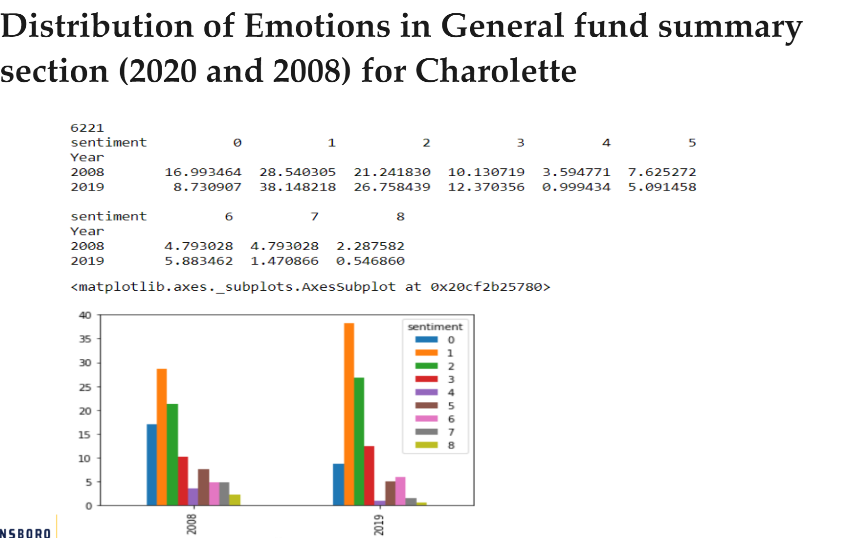
**Emotional and sentiment analysis**

1. **Data exploration and statistical evaluation of data:**

Below picture shows most influential words in Guilford county budget document in 2008 and 2020. For example, public was the most influential word in 2020 but in 2008 during recession emergency word was used the most.







Time taken to perform the task: Around 4 to 5 hours

Link to the Ipython notebook : <https://github.com/UNCG-CSE/Budget_Text_Analysis/blob/master/src/EmotionAnalysis/Budget_Analysis_Emotions1.ipynb>

Results explained:

The graph above shows that difference of emotions between 2008 and 2019. This experiment was to show that there was a lot of difference in emotions in 2008 (During Recession) compared to 2019. The graph shows that positivity has increased in 2019 and negativity has decreased.

1. **Hypothesis Testing:**

H0 -> The sentiments remain same for service part from 2008 and 2020 in Charlotte city document.

H1 -> Sentiment changes for service part from 2008 to 2020 in charlotte city document

To prove this Hypothesis two sample is performed and p-value threshold is p = 0.05

Time taken to perform the task: Around 2 hours

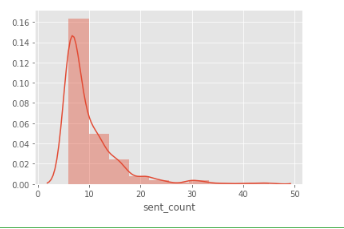
Result:

After applying the test, we found out that P-Value is greater than threshold (0.56) therefore we were failed to reject null hypothesis.

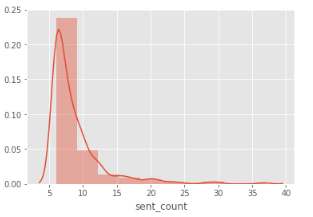
1. **Probability Distribution:**

I have concatenated Guilford county, Durham county, Durham city, charlotte city, Raleigh city:

Took negative sentiment counts (at least more than 5 times)



Took positive sentiment counts (at least more than 5 times):



Time taken to perform the task: Around 2 hours

Link to the Ipython notebook : <https://github.com/UNCG-CSE/Budget_Text_Analysis/blob/master/src/EmotionAnalysis/Combined%20Distribution.ipynb>

Result:

I applied K-test and found out that the p-value is 1.0 for poisson distribution, it means Poisson distribution is fitting well in this.

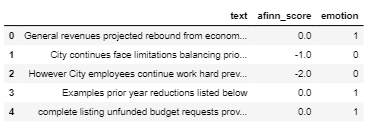
1. **Machine Learning:**

**Reforming data and data exploration:**

Data given to us was in single word therefore it was not making sense for us to apply machine learning on top of it. I have parsed the data again and reformed it in a data frame. Here are the steps used to reform the data:

1. Parsed the pdf file with the help of Tika parser.
2. Converted it to string
3. Broke the data into sentences
4. Data cleaning (Removing unwanted characters, Tokenization, Removing stop words, Lemmatizing/Stemming.)
5. Dropped the rows which are empty
6. Used Affin library from python to assign Affin values
7. Assigned the sentiments accordingly

New data frame looks like:



Question: My data is charlotte city funds for 2008 where according to sentiment analysis negativity is higher. We are classifying how accurately it is classifying sentiments in the document.

X = Vectorized text (vectors)

Y = emotions

Used This vectorizer which breaks text into single words and bi-grams and then calculates the TF-IDF representation.



Time taken to perform the task: Around 10 hours

Link to the Ipython notebook <https://github.com/UNCG-CSE/Budget_Text_Analysis/blob/master/src/EmotionAnalysis/Combined_Machine_%20Learning.ipynb>

Results explained:

Here 91% times classifier was able to classify the emotions correctly and root mean square error is 0.28.