Budget Text Analysis

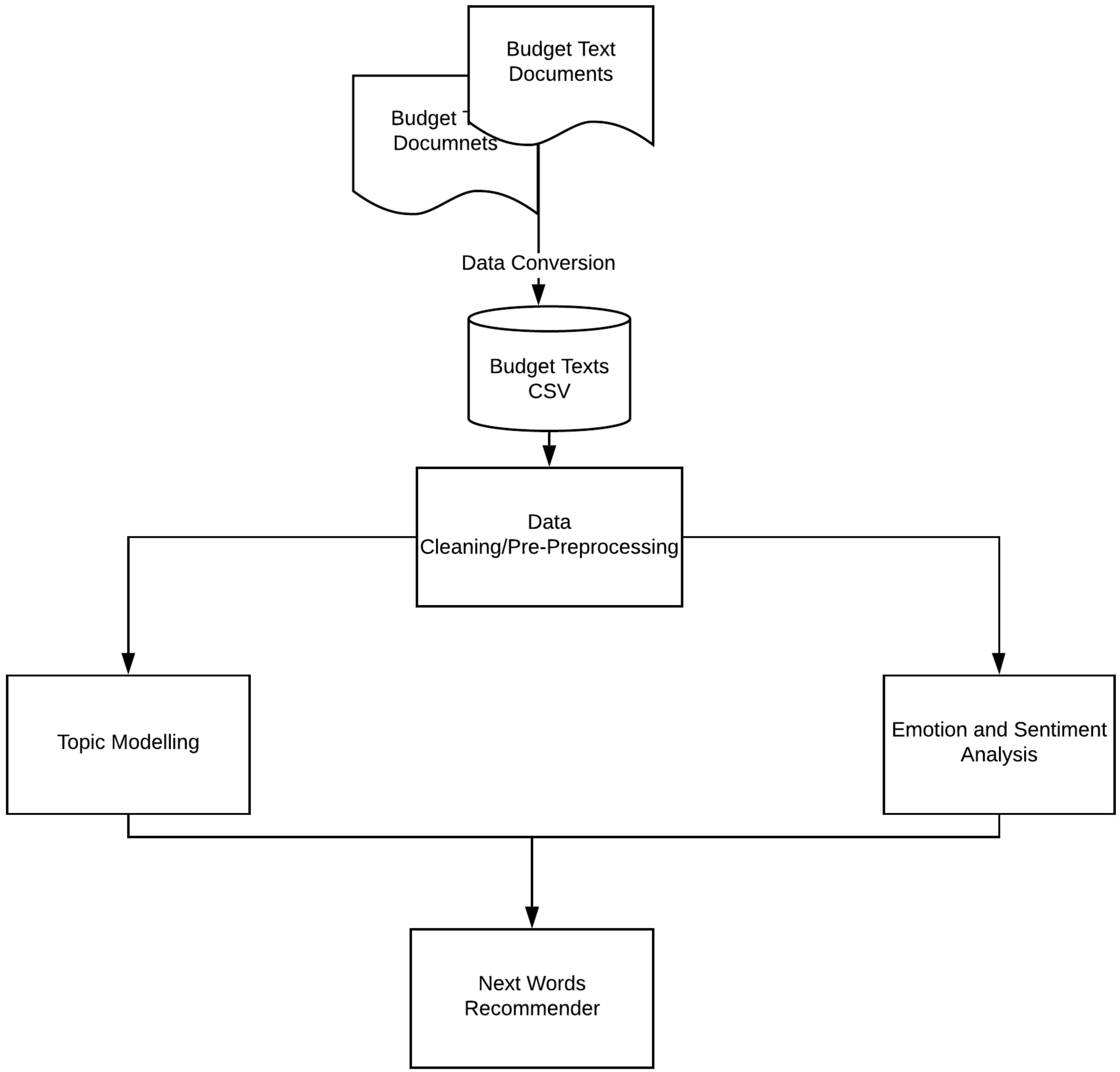
- Datatopian Visionaries

Akash Meghani,
Miguel Gaspar Utrera,
Naseeb Thapaliya,
Sultan Al Bogami,
Unnati Khivasara

Mentors: Dr. Soumya Mohanty
Jason Jones (Guilford County)

UNC GREENSBORO

Overview of the Project



Goals (Questions Formulated)

- **What are the properties of the budget text data ?**
- **Are there any intrusive structures and patterns in the budget texts from all the counties ?**
- **Does the topics change over the years (2008, 2029) ?**
- **Does different sections of budget text data have any common relation between them ?**
- **Does general funds section of Guilford county, Durham County and Charlotte City give similar sentiments or they are different ?**
- **How can we visualize the topics and emotions between 2008 and 2019 with proper analytics ?**
- **Does a topic model for one year can identify the latent semantic structure that persists over time in this budget text domain ?**
- **Can we formulate a next word recommender from our analysis ?**

Research Objectives

- **Understand the Different sections of budget text data from different counties and create a relation between them.**
- **Compare the general funds section of Guilford county, Durham County and Charlotte City (2008 and 2019) and understand the difference between them.**
- **Visualization of emotions between 2008 and 2019.**
- **Understand the different relevant topics from all the counties 2019 and with computed their coherence score with proper visualization.**
- **Compared the topic modeling results over the years (2008,2020)**
- **Perform supervised machine learning with the topics obtained from topic modeling as input and apply it to other years.**
- **Build a simple next word recommender based on the topic modeling.**

Tasks Assigned

- Sultan Al Bogami
  1. Collected Budget Documents from all the different Counties websites and other sources(2008 to 2020) and organization of github.
  2. Converted the pdf documents to ccv formats. Extract words from the documents using online tool, and classify them for further processing.
  3. Perform Statistical analysis of budget texts
- Naseeb Thapaliya
  1. Compared the topic modeling results over the years (2008,2020)
  2. Perform Supervised Machine Learning on topics from topics Modeling.
- Miguel Gasper Utrera
  1. Applied Topic modeling on different relevant topics from all the counties and computed their coherence score with proper visualization.
- 2. Applied Davis model and showed top 30 words in each topic and their relevence.
- Unnati Khivasera
  1. Analyzing sentiment intensity using Vader.
  2. Performed visualization of emotions from different sections of documents.
- Akash Meghani
  1. Applied Emotional and Sentiment analysis with NLTK and got meaningful results.
  2. Performed visualization of emotions from different sections of documents.
- Everyone
  1. Documentation of project and maintain and work on GitHub.
  2. Work on Nextword recommender

Data Overview

- **Primarily, 7 pdf files ranging from 400-500 pages long for each.**
- **Each pdf is converted to csv files by extracting all the relevant budget texts(words) from the pdf file.**
- **So, there are total of 638131 total words extracted from the budget files.**

Data Source

Guilford County
STATE of NORTH CAROLINA

Search...

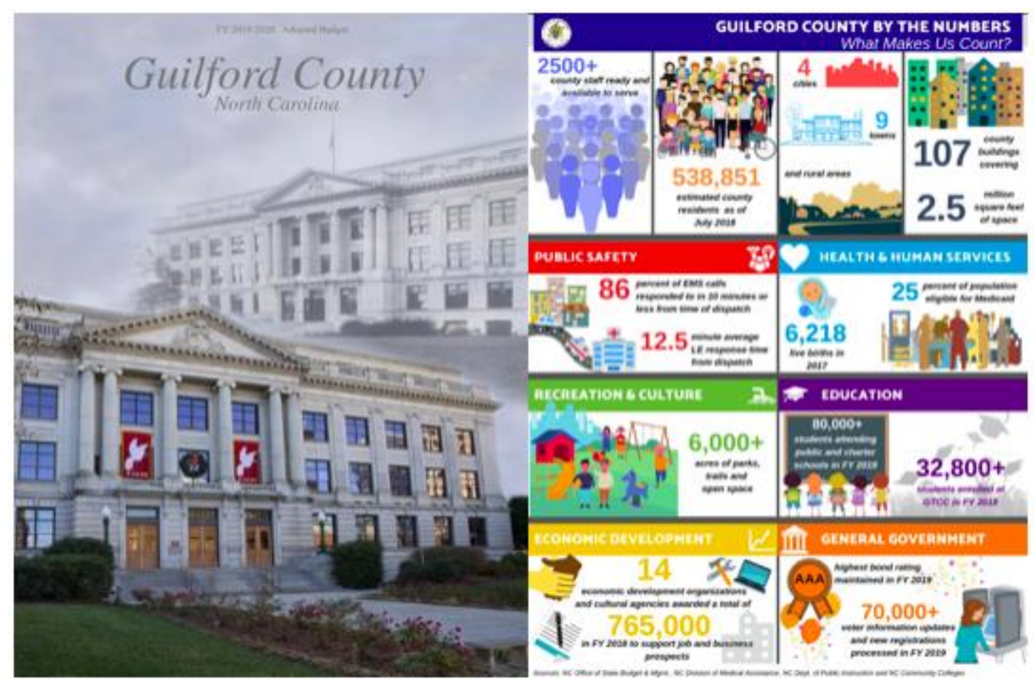
Services | Our County | Business | Get Connected | How Do I...

- Budget, Management & Evaluation
  - FY 2019-20 Adopted Budget
  - How are your Tax Dollars Spent?
  - Budget Amendments Reports
  - Budget Performance Reports
  - + Budget History & Past Adopted Budget Documents
  - + Capital Investment Plan & Capital Project Status
  - Other Financial Information
  - Contact Information

Our County » Budget, Management & Evaluation »

FY 2019-20 Adopted Budget

Font Size: Share & Bookmark Feedback Print



FY 2019-20 Adopted Budget Document

FY 2019-20 Adopted Budget-in-Brief

Data Analysis

In [47]:
```
Combined_df.shape
```
Out[47]:
```
(638131, 3)
```

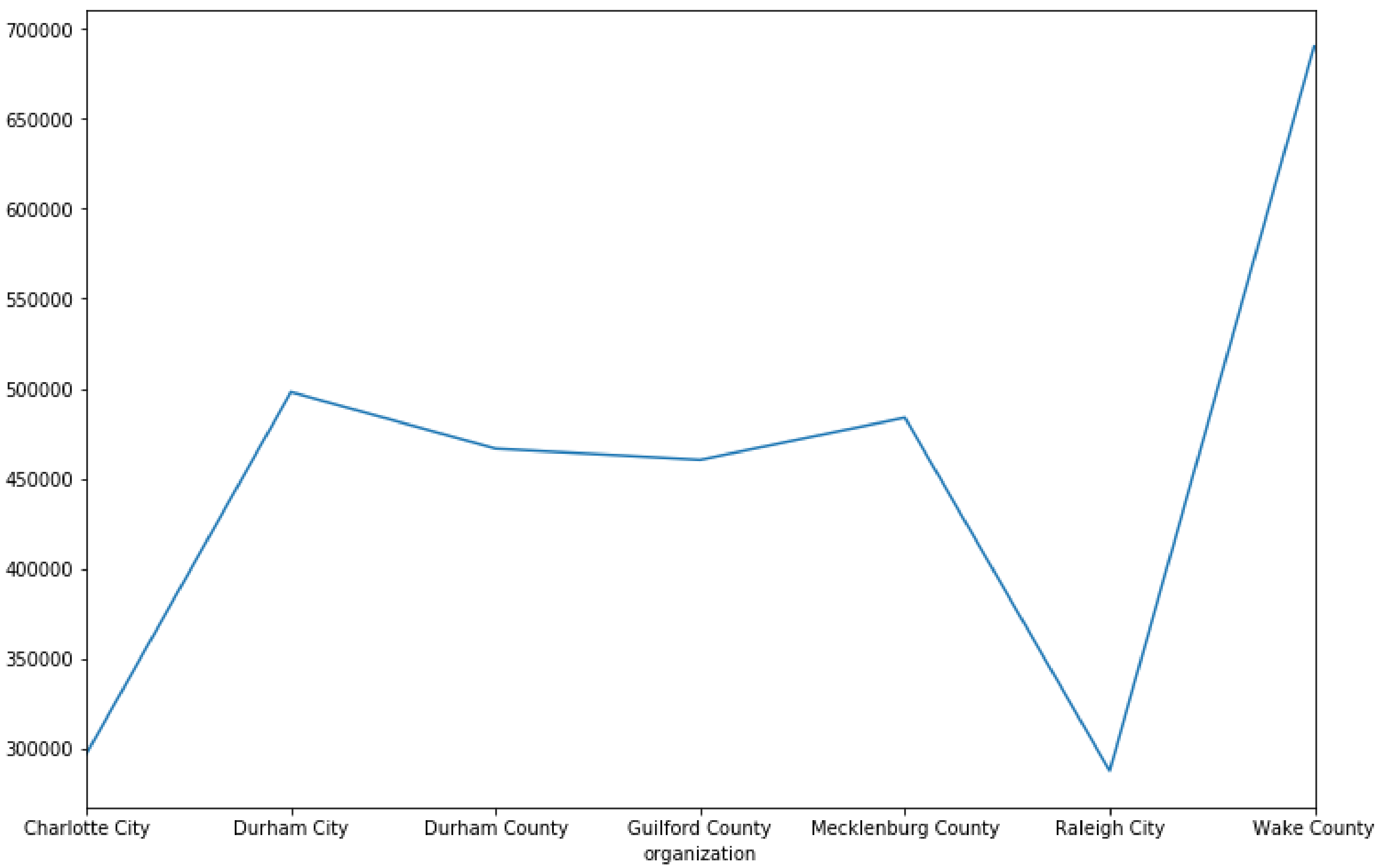
In [45]:
```
Combined_df.describe()
```
Out[45]:

	page_number
count	638131.000000
mean	213.602262
std	137.058241
min	1.000000
25%	100.000000
50%	203.000000
75%	305.000000
max	537.000000

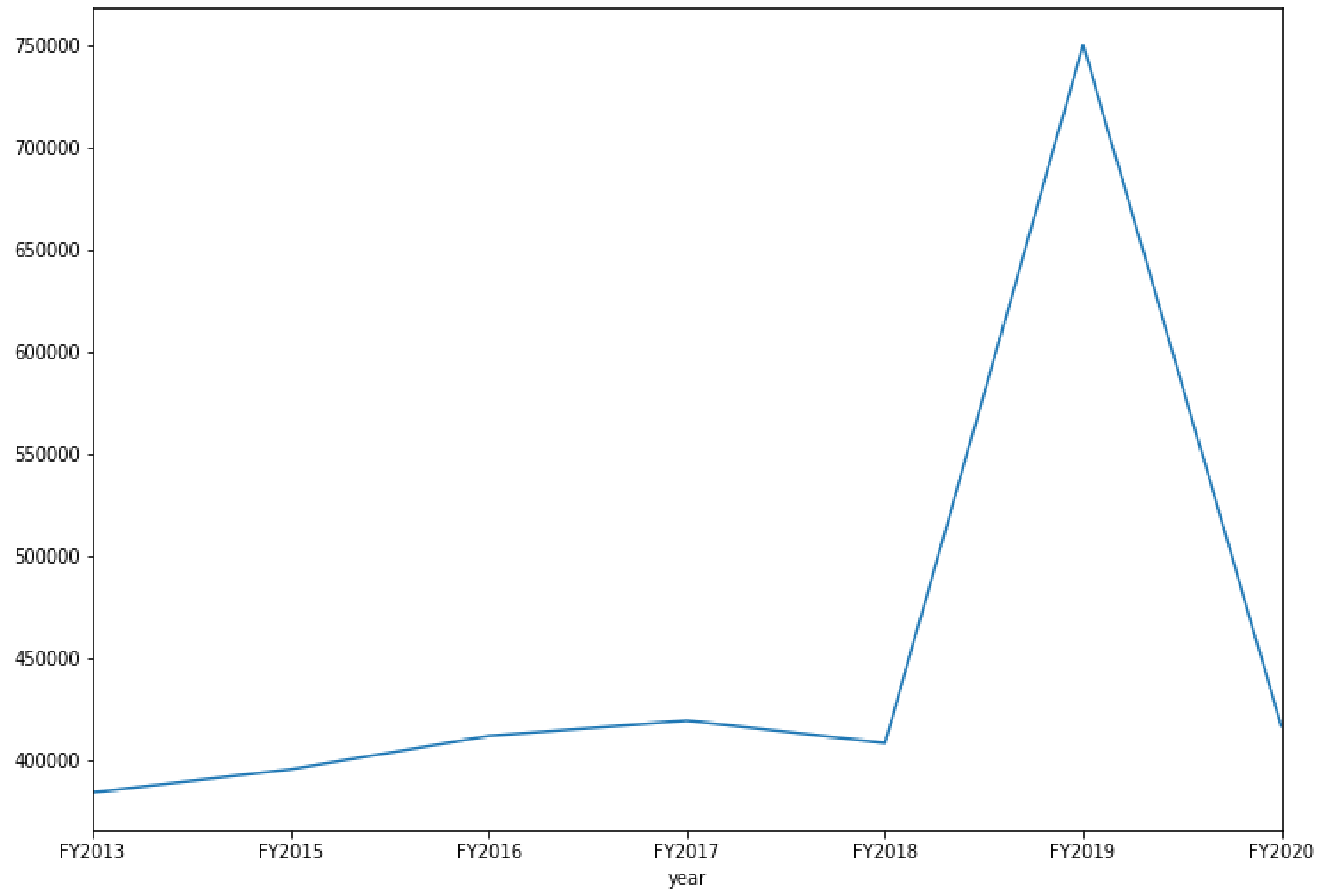
In [50]:
```
Combined_df.to_csv("Combined_Counties.csv", sep='\t', encoding='utf-8')
```

In [ ]:

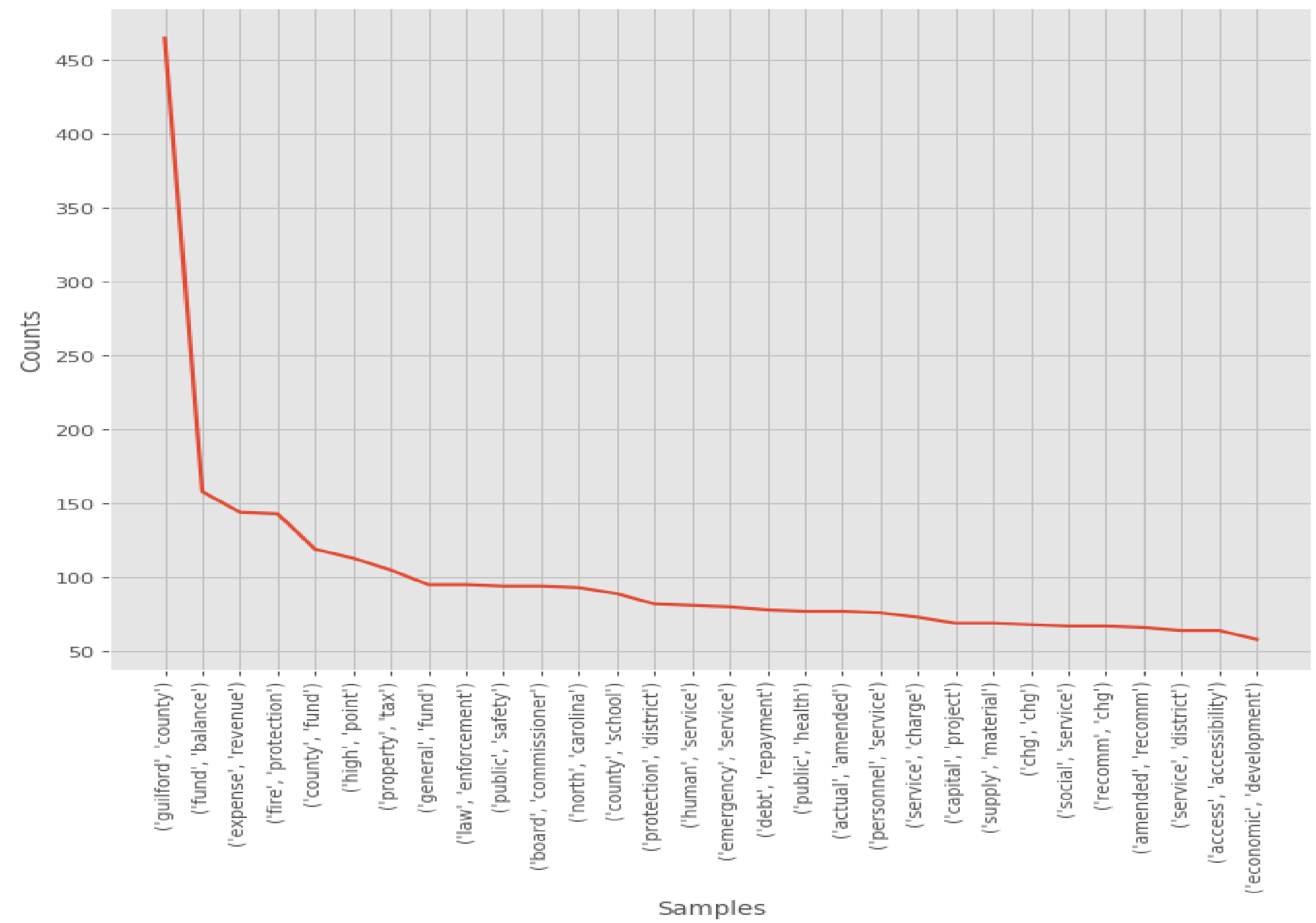
Count of words grouped by organizations.



Count of words grouped by year.



Most Frequent bigrams in Guilford County budget document From 2020



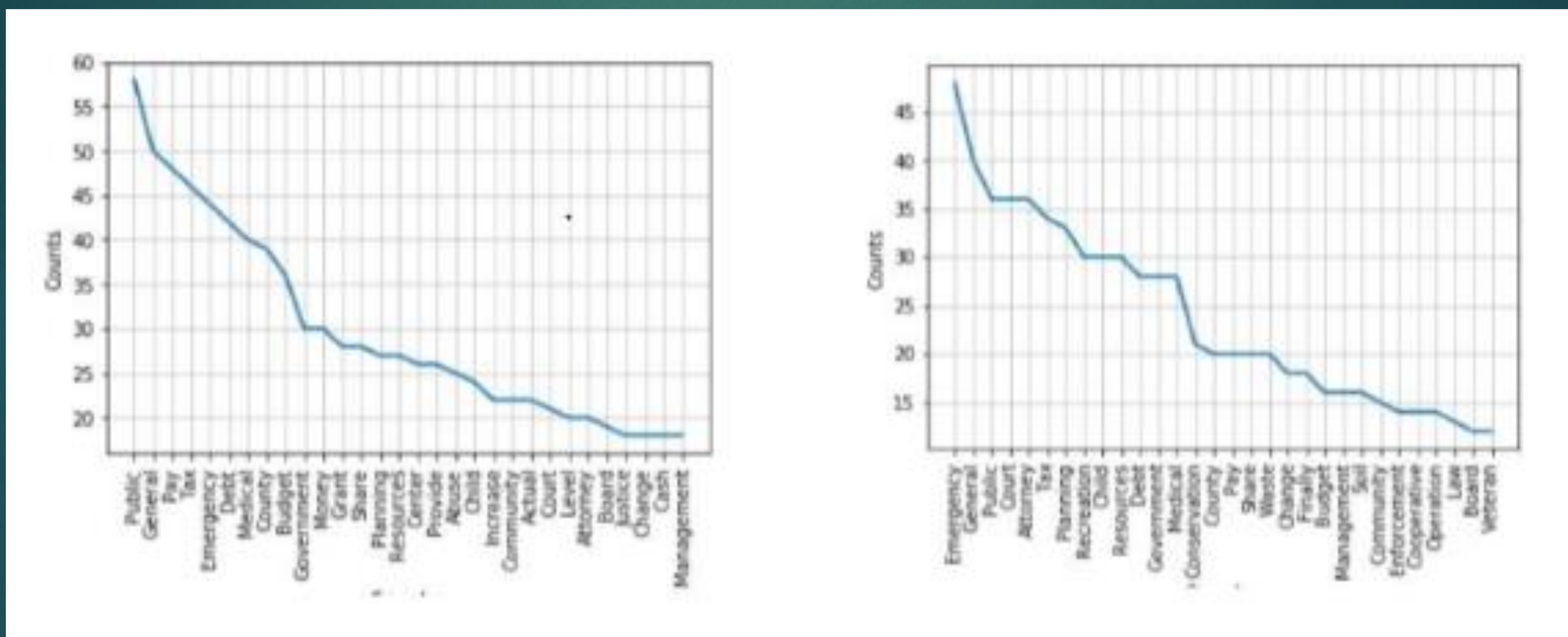
Sentiment Analysis : Influential words

Most Influential Words in Guilford County (2020 and 2008)

```
[('Public', 58),
 ('General', 50),
 ('Pay', 48),
 ('Tax', 46),
 ('Emergency', 44),
 ('Debt', 42),
 ('Medical', 40),
 ('County', 39),
 ('Budget', 36),
 ('Government', 30)]
```

```
[('Emergency', 48),
 ('General', 40),
 ('Public', 36),
 ('Court', 36),
 ('Attorney', 36),
 ('Tax', 34),
 ('Planning', 33),
 ('Recreation', 30),
 ('Child', 30),
 ('Resources', 30)]
```

Frequency of most influential words(2020 and 2008)



Sentiment Analysis : Sentiment renaming

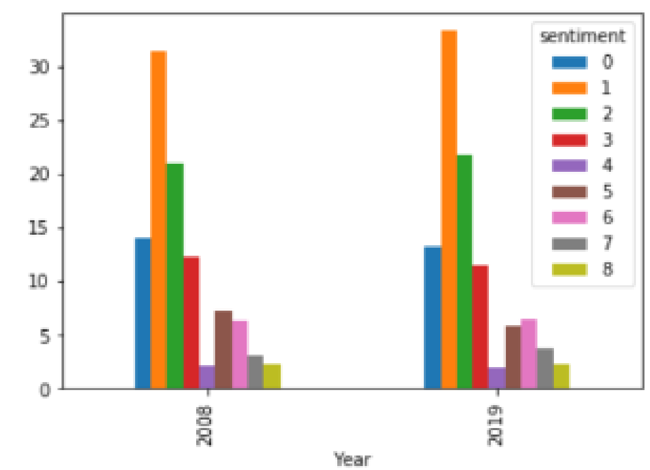
"Negative": "0","Positive":
"1","Trust" :"2","Sadness":"0","Anticipation":"3","Surprise":"4","Fear":"5","Joy":"6","Anger":"7","Disgust":"8"

Distribution of Emotions Services section for Guilford County

Distribution of Emotions Services section for Guilford County

sentiment Year	0	1	2	3	4	5	6	7	8
2008	13.982430	31.442167	20.973646	12.262079	2.159590	7.320644	6.368960	3.111274	2.379209
2019	13.250518	33.258046	21.682665	11.424807	2.051572	5.853567	6.437041	3.707886	2.333898

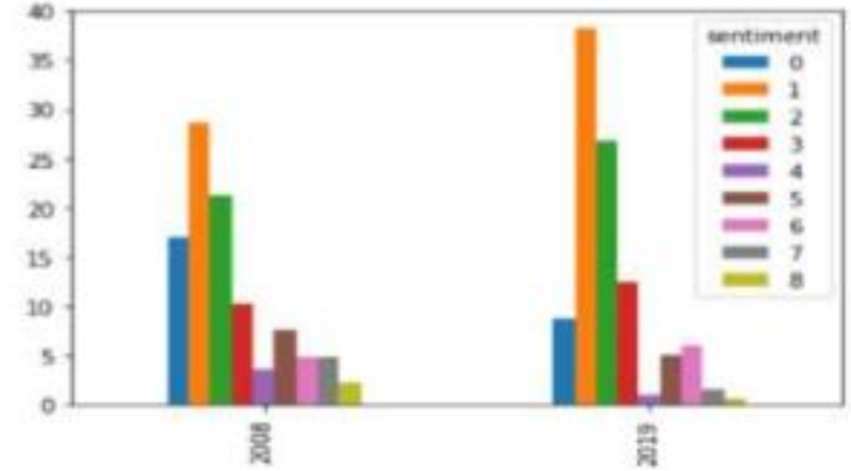
```
Out[23]: <matplotlib.axes._subplots.AxesSubplot at 0x1dc873ac828>
```



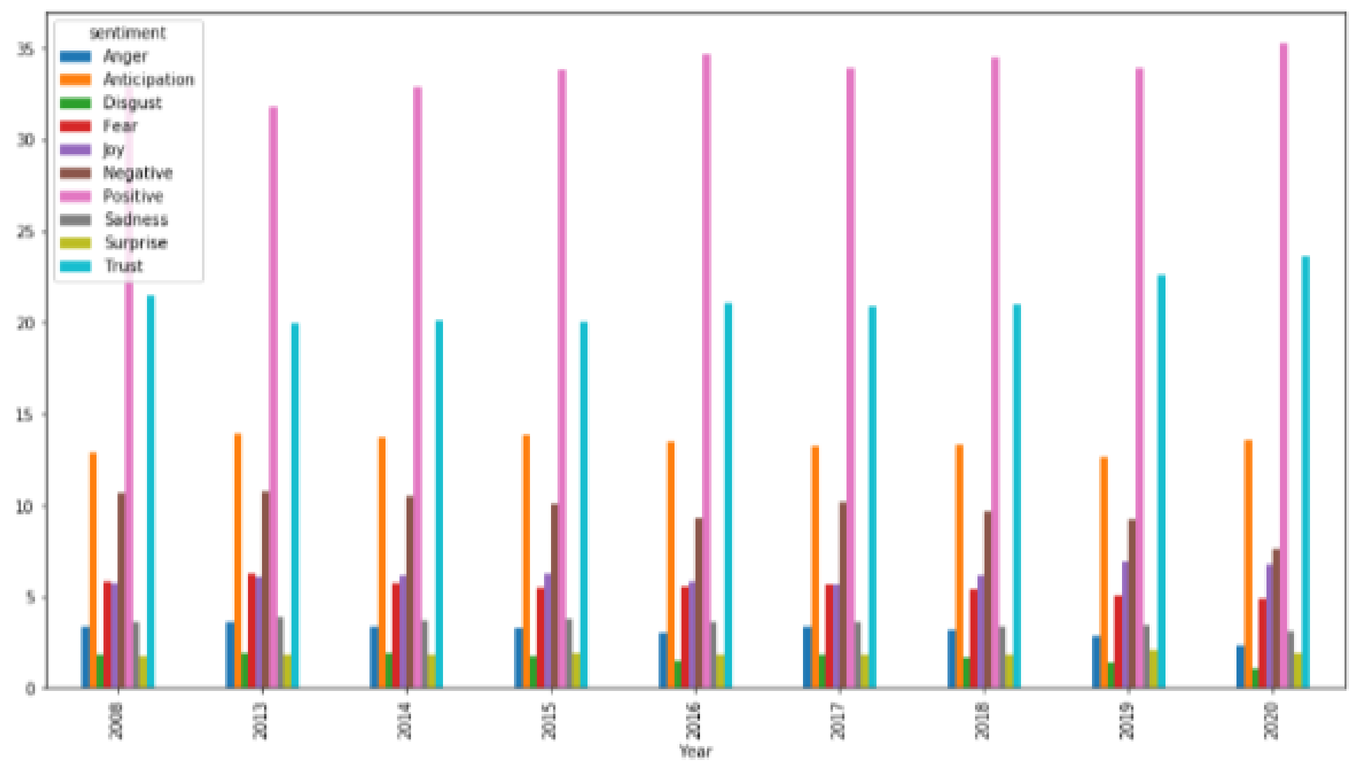
```
6221
sentiment          0          1          2          3         4         5
Year
2008       16.993464  28.540305  21.241830  10.130719  3.594771  7.625272
2019        8.730907  38.148218  26.758439  12.370356  0.999434  5.091458

sentiment         6         7         8
Year
2008       4.793028  4.793028  2.287582
2019       5.883462  1.470866  0.546860

<matplotlib.axes._subplots.AxesSubplot at 0x20cf2b25780>
```



Distribution of Emotions in General Fund section (2008 and 2020) for Charlotte County



Charlotte sentiments and emotions distribution over the years(2008 and 2013 to 2020):

Charlotte Sentiment Continue..

- The plot shows that Positive sentiments increased after 2008 till year 2016 and slightly dropped in 2017 and remained stable in further years.
- While the Negative sentiments have reverse impact, as they dropped till year 2016 and increased in 2017 and then again dropped till 2020.
- Also the emotions like Disgust and Fear kept reducing over the years while Anticipation remained almost same for all years.

Hypothesis Testing:

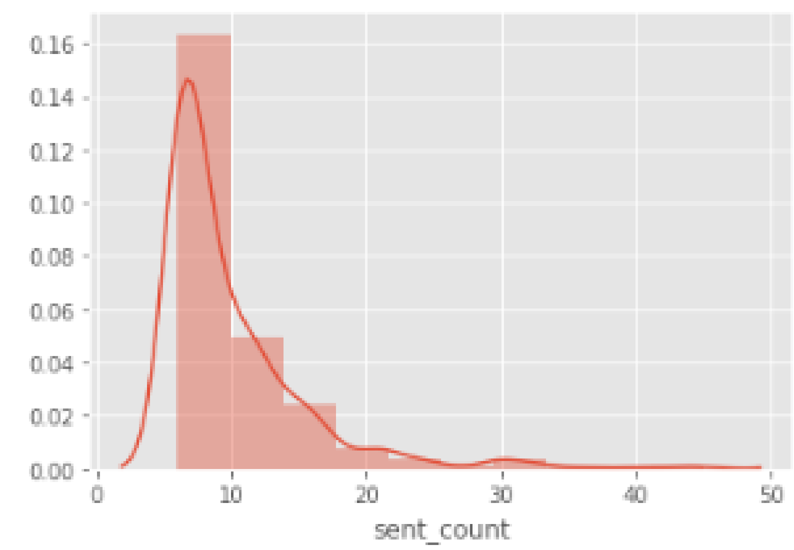
- H0 -> The sentiments remain same for service part from 2008 and 2020.

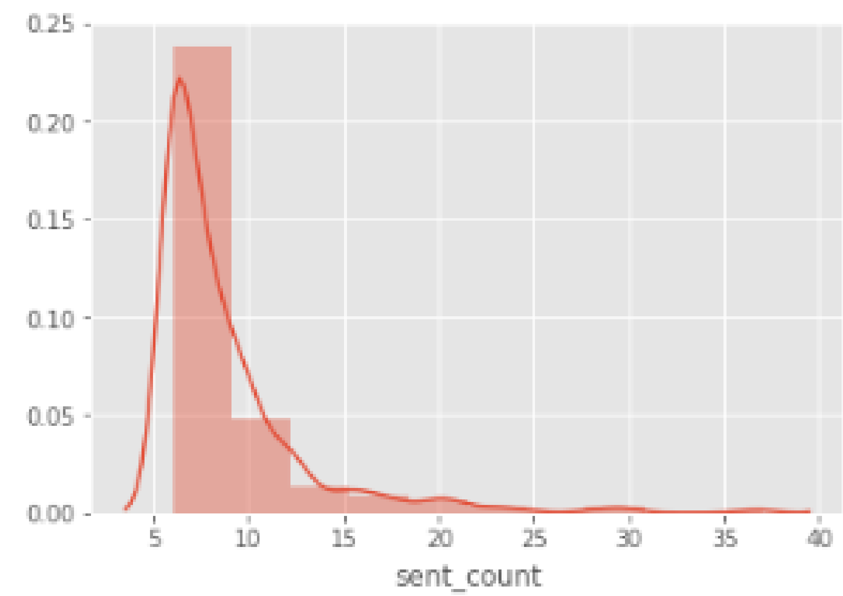
  H1 -> Sentiment changes for service part from 2008 to 2020.

- To prove this Hypothesis two sample is performed and p-value threshold is p = 0.05
- P-Value is greater than threshold (0.56) therefore we were failed to reject null hypothesis.

Probability Distribution:

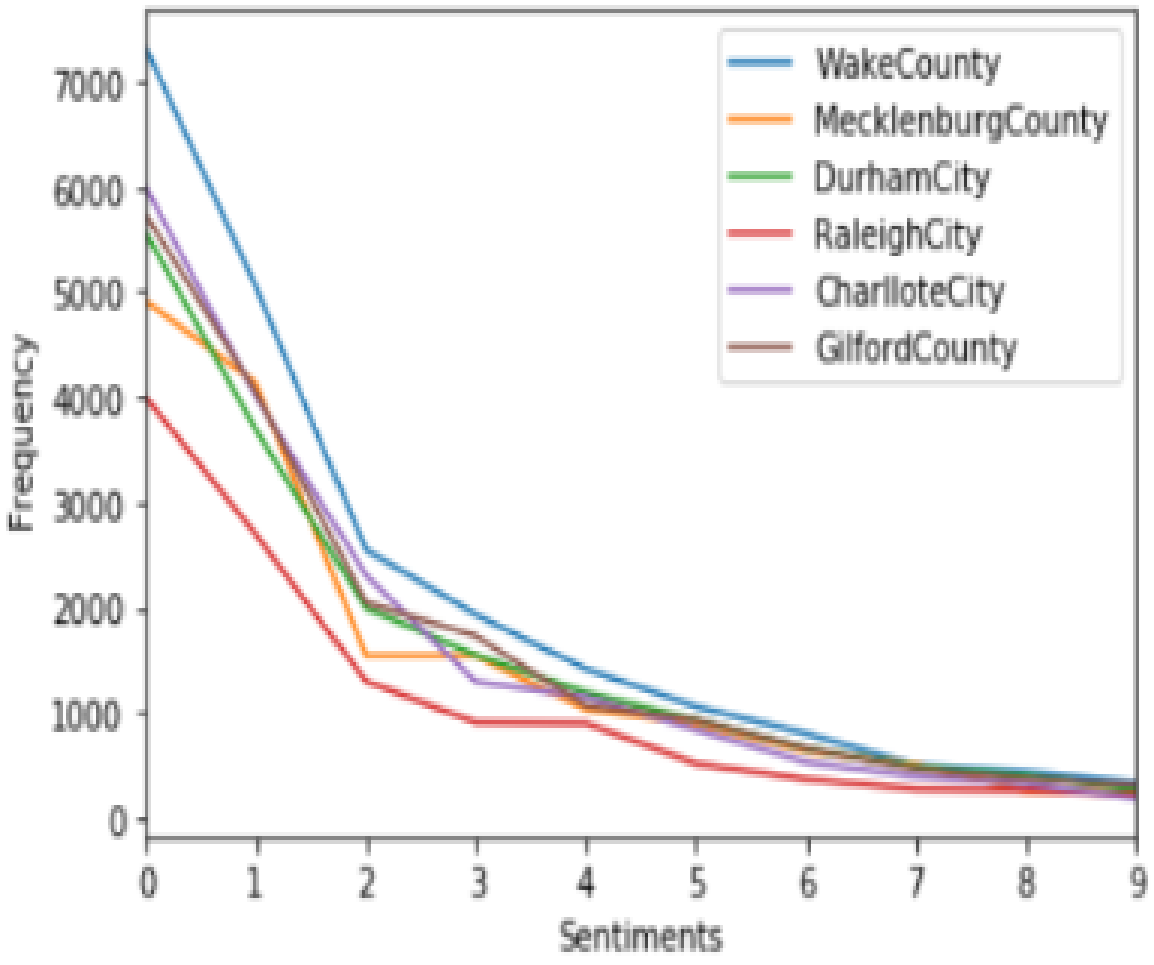
- I have concatenated Guilford county, Durham county, Durham city, charlotte city, Raleigh city :
- Took negative sentiment counts (at least more than 5 times).
- Took positive sentiment counts (at least more than 5 times):

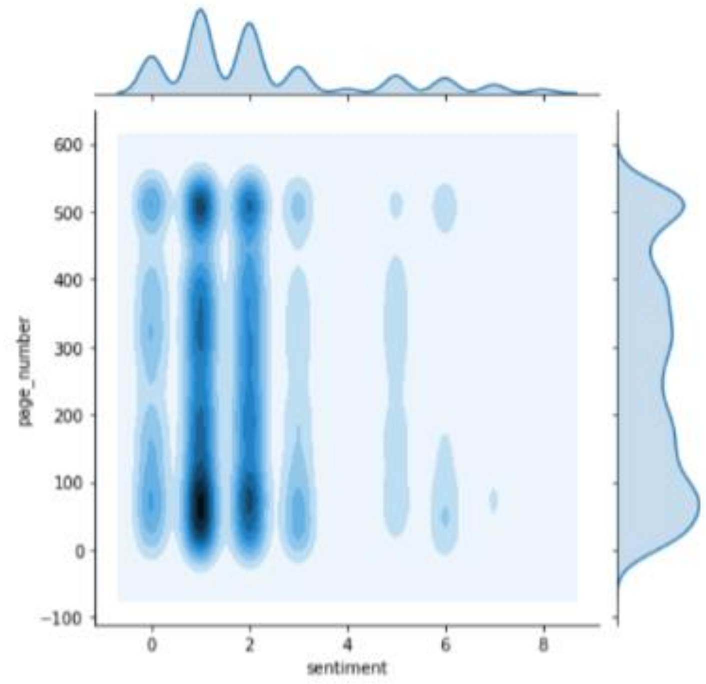


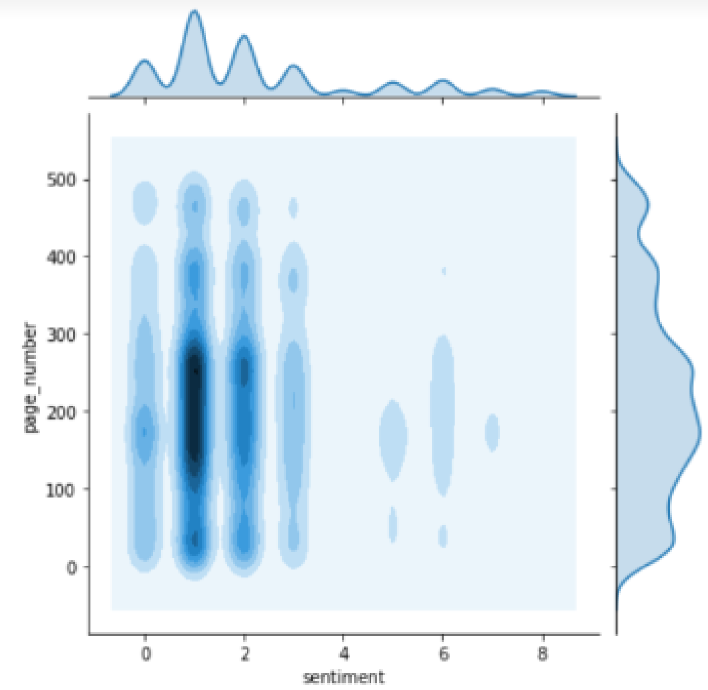


sentiments and emotions for all the cities

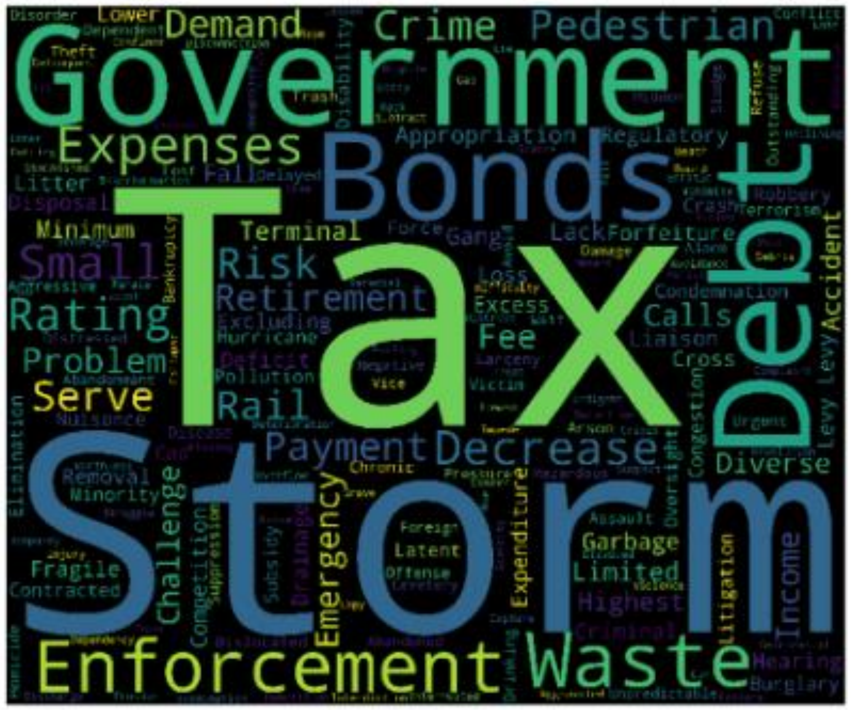
- Frequency Distribution of sentiment and emotions in the budget document remains the same.
- In Mecklenburg county it is noticed that the negative sentiment is slightly increased while this is not seen for all the cities.







Sentiment and emotion distribution with respect to page number



Charlotte city negative words (2008 and 2020)

Machine Learning

- Sample data

	text	afinn_score	emotion
0	General revenues projected rebound from econom...	0.0	1
1	City continues face limitations balancing prio...	-1.0	0
2	However City employees continue work hard prev...	-2.0	0
3	Examples prior year reductions listed below	0.0	1
4	complete listing unfunded budget requests prov...	0.0	1

Machine learning

- Split the data in 70/30 for creating train/test dataset.
- TF-IDF was used on training data. This vectorizer breaks text into single words and bi grams and create TF-IDF representation to create feature vectors.
- X -> Vectorized text
- Y-> (positive, negative)

Machine Learning

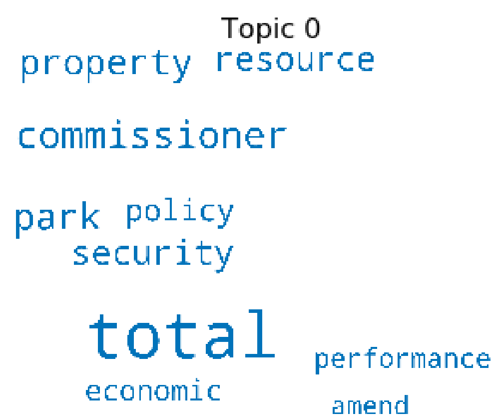
- Results

```
<class 'scipy.sparse.csr.csr_matrix'>
RMSE : 0.41633319989322654
Accuracy : 82.67%
RMSE : 0.3651483716701107
Accuracy : 86.67%
RMSE : 0.32659863237109044
Accuracy : 89.33%
```

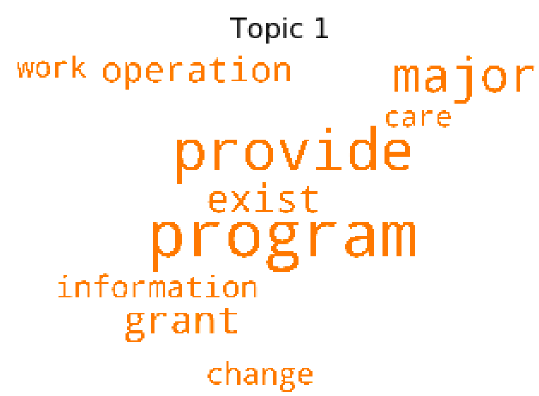
Topic Modeling

```
[(0,
  '0.315*"total" + 0.056*"commissioner" + 0.052*"park" + 0.051*"property" + '
  '0.044*"security" + 0.044*"resource" + 0.035*"policy" + 0.032*"economic" + '
  '0.027*"performance" + 0.026*"amend"'),
 (1,
  '0.196*"program" + 0.153*"provide" + 0.106*"major" + 0.064*"grant" + '
  '0.063*"exist" + 0.053*"operation" + 0.039*"information" + 0.037*"change" + '
  '0.035*"work" + 0.034*"care"'),
 (2,
  '0.115*"fund" + 0.110*"summary" + 0.108*"fire" + 0.078*"area" + '
  '0.062*"current" + 0.060*"solid" + 0.048*"state" + 0.041*"level" + '
  '0.040*"percent" + 0.039*"estimate"'),
 (3,
  '0.187*"fiscal" + 0.086*"debt" + 0.074*"unit" + 0.068*"water" + '
  '0.060*"infrastructure" + 0.050*"issue" + 0.044*"goal" + 0.042*"remain" + '
  '0.042*"government" + 0.041*"base"'),
 (4,
  '0.206*"adopt" + 0.107*"replacement" + 0.090*"support" + 0.083*"increase" + '
  '0.075*"number" + 0.044*"charge" + 0.041*"planning" + 0.038*"additional" + '
  '0.038*"require" + 0.038*"site"'),
 (5,
  '0.219*"capital" + 0.134*"expenditure" + 0.100*"management" + '
  '0.072*"equipment" + 0.050*"balance" + 0.044*"vehicle" + 0.040*"begin" + '
  '0.036*"improve" + 0.030*"identify" + 0.026*"law"'),
 (6,
  '0.242*"include" + 0.164*"community" + 0.095*"school" + 0.075*"impact" + '
  '0.037*"rate" + 0.033*"maintain" + 0.027*"recommend" + 0.027*"associate" + '
  '0.026*"pay" + 0.024*"resident"'),
 (7,
  '0.259*"year" + 0.150*"funding" + 0.117*"public" + 0.080*"development" + '
  '0.077*"actual" + 0.044*"plan" + 0.029*"annual" + 0.024*"life" + '
  '0.021*"address" + 0.019*"help"'),
 (8,
  '0.047*"service" + 0.019*"system" + 0.013*"building" + 0.012*"improvement" + '
  '0.012*"operate" + 0.011*"transfer" + 0.010*"cost" + 0.010*"source" + '
  '0.010*"complete" + 0.009*"future"'),
 (9,
  '0.260*"budget" + 0.207*"project" + 0.180*"facility" + 0.133*"revenue" + '
  '0.052*"tax" + 0.024*"appropriate" + 0.024*"control" + 0.015*"specific" + '
  '0.014*"population" + 0.011*"food"')]
```

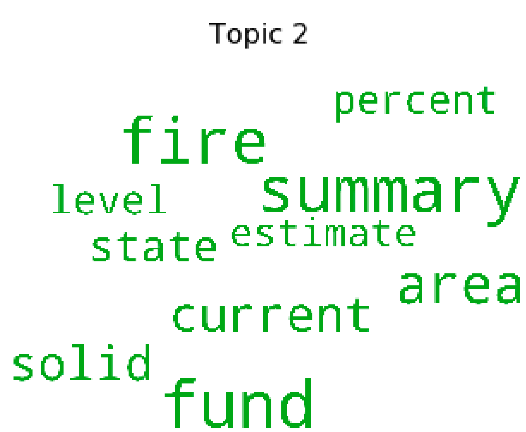
Topic Modeling



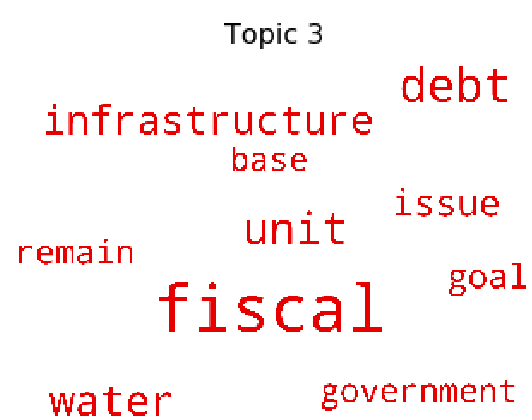
Label: Property Maintenance and Security



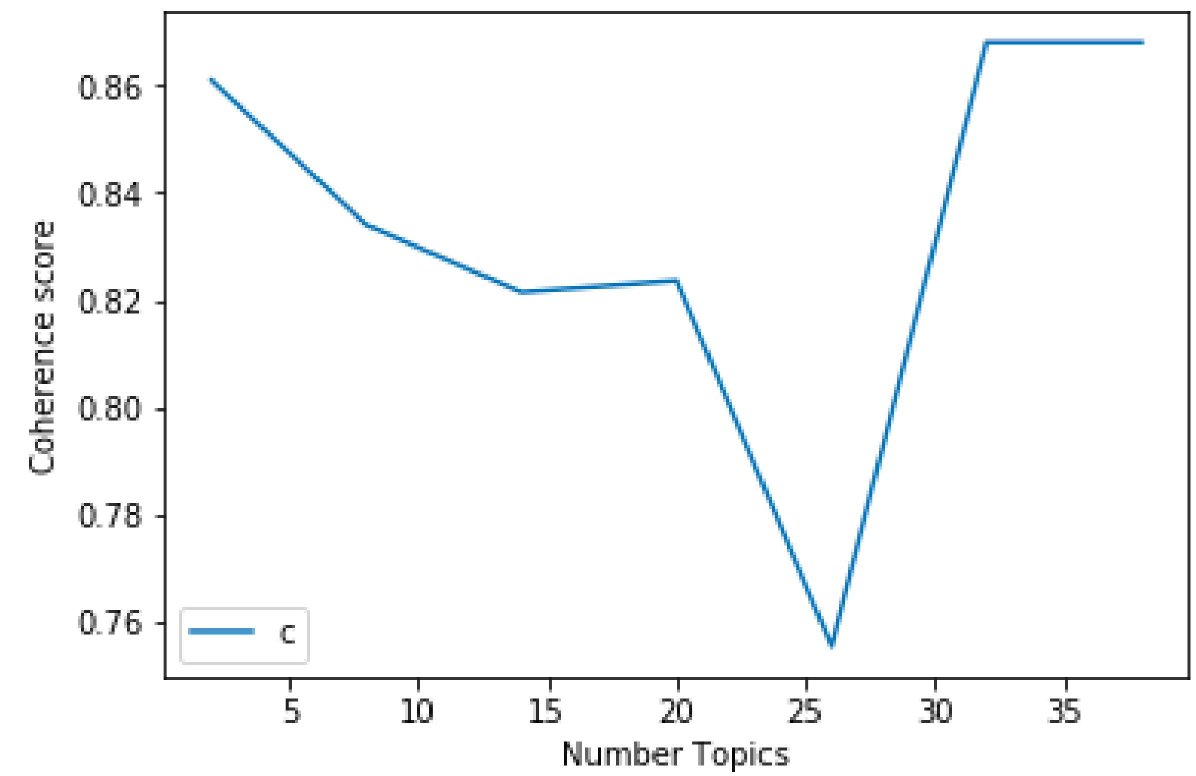
Label: Grant for Work or Program



Label: State Fire Fund

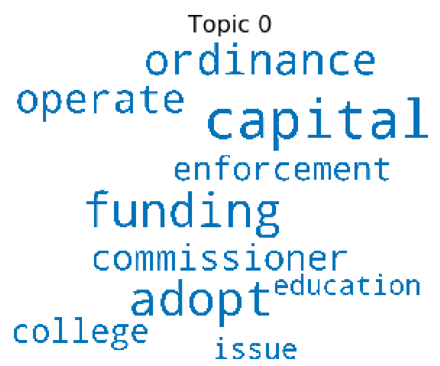


Label: Government Fiscal Year

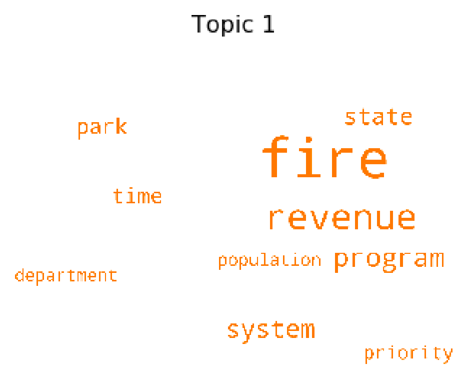


Coherence Score: 0.8256146597574272

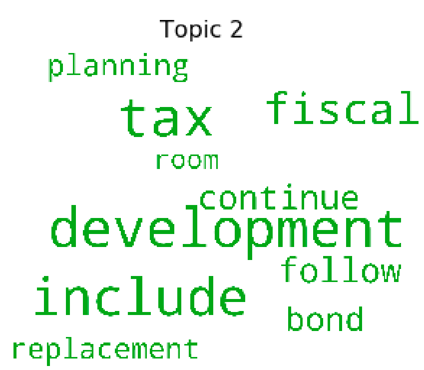
Topic Modeling Comparison



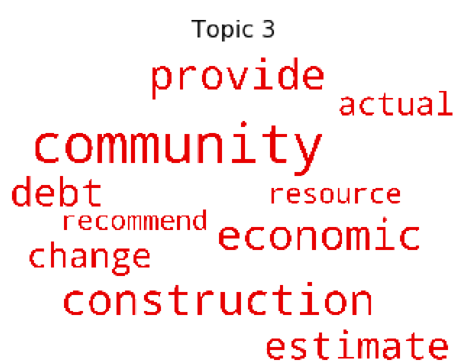
Label: Capital Funding for Education



Label: Fire Fund Program

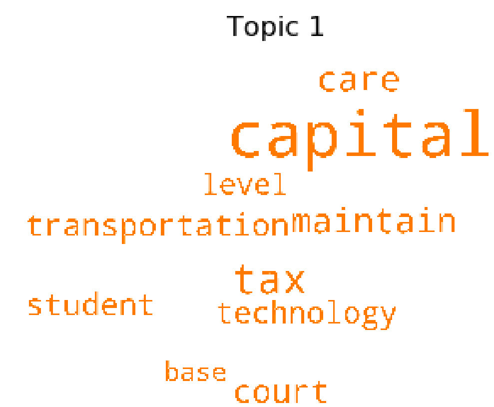


Label: Fiscal Year Planning

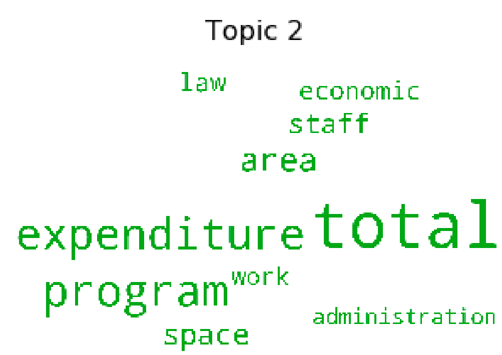


Label: Community Construction

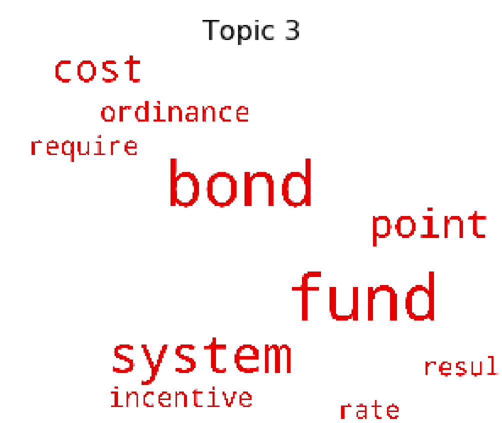
Label: Increase risk in project, department



Label: Student Tax



Label: Economic Expenditure



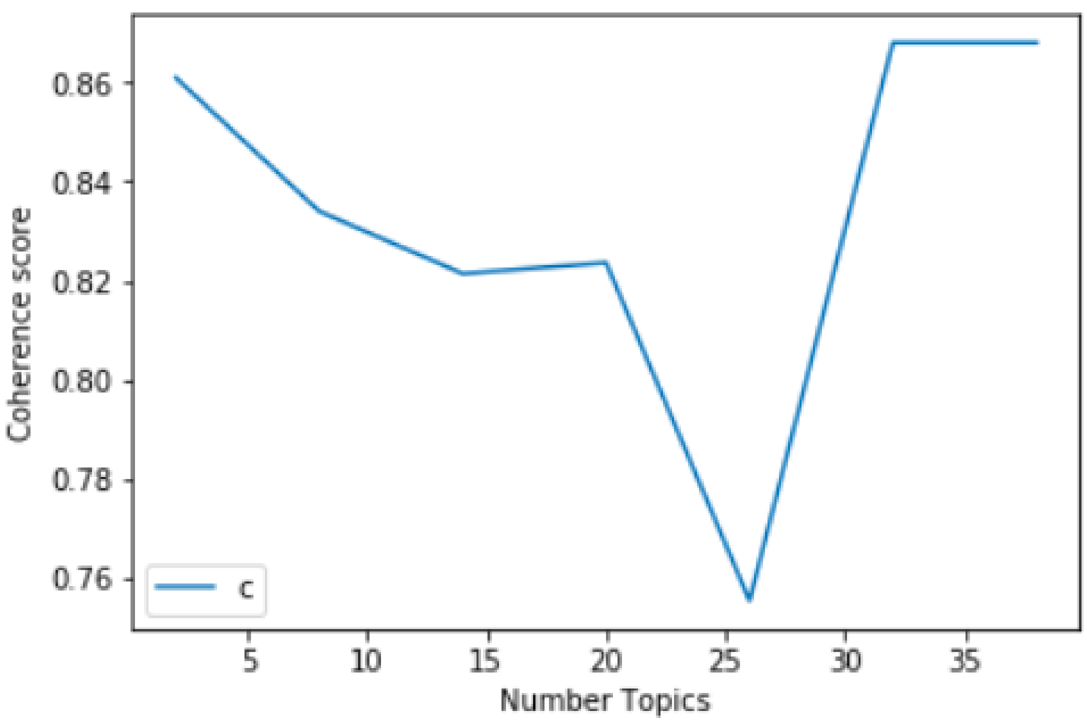
Label: Financial Indicators

2019

2008

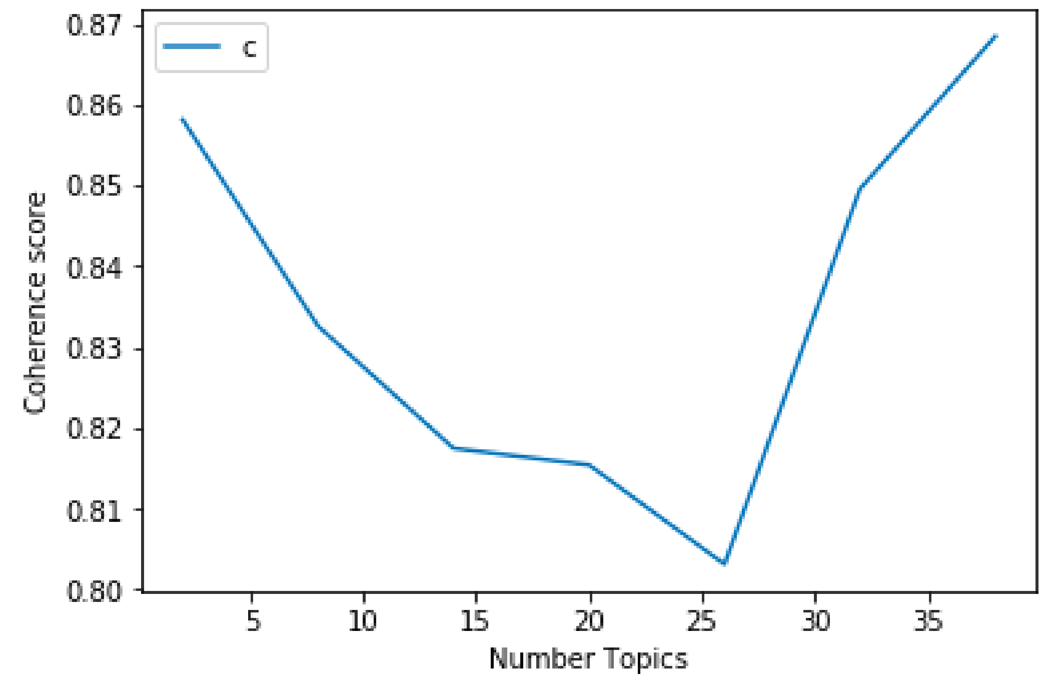
UNC GREENSBORO

Topic Modeling Comparison




```
Coherence Score:  0.8256146597574272
```

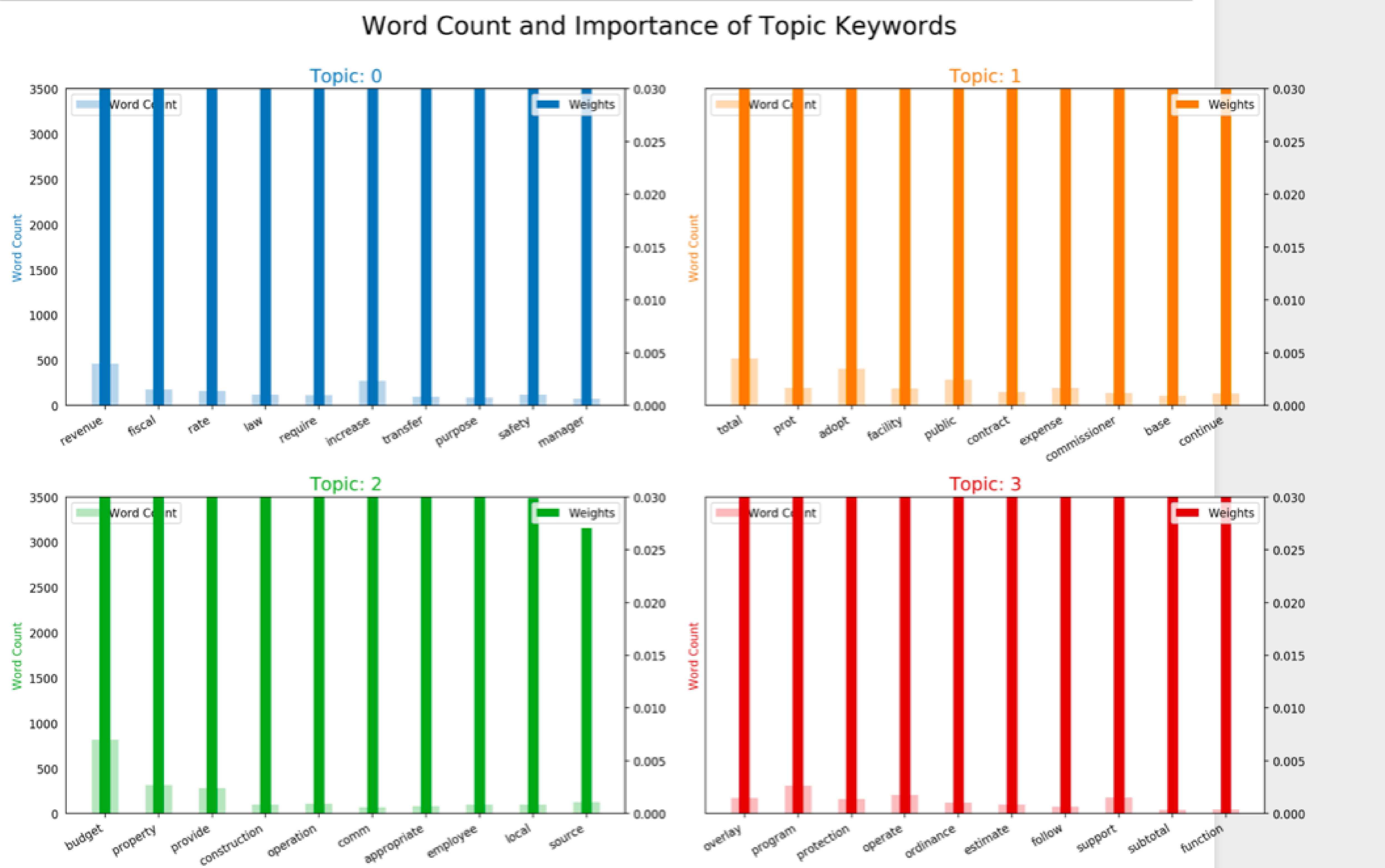
2019




```
Coherence Score:  0.8247949042506306
```

2008

Topic Modeling



Tasks

- Train LDA Model on the budget texts from 2019.
- Grab Topic distributions for every budget texts using the LDA Model
- Use Topic Distributions directly as feature vectors in supervised classification models (Logistic Regression, SVM, etc) and get F1-score.
- Use the same 2019 LDA model to get topic distributions from 2018 and 2020 (**the LDA model did not see this data!**)
- Run supervised classification models again on the 2018 and 2020 vectors and see if this generalizes.

Converting Topics to Feature Vectors for Machine Learning

In [108]:
```
train_vecs = []
for i in range(len(GC_df)):
    top_topics = lda_model.get_document_topics(corpus[i], minimum_probability=0.0)
    topic_vec = [top_topics[i][1] for i in range(10)]
    topic_vec.extend([GC_df.iloc[i].sent_count]) # counts of reviews for restaurant
    topic_vec.extend([len(GC_df.iloc[i].word)]) # length review
    train_vecs.append(topic_vec)
```

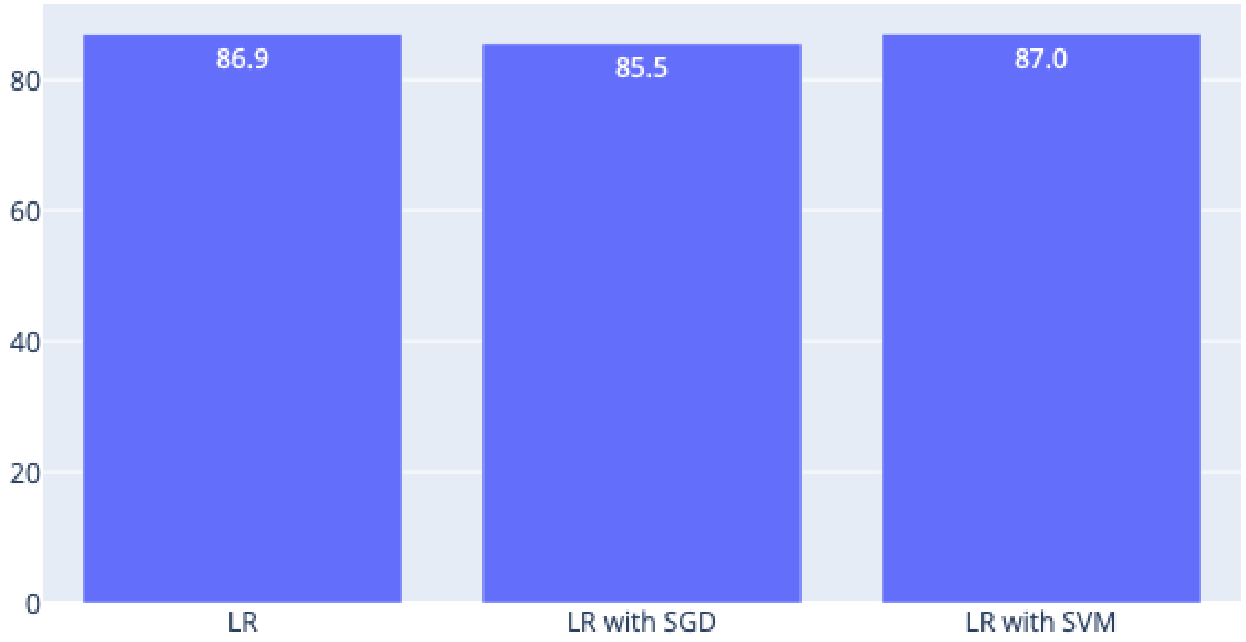
In [109]:
```
train_vecs[2]
```

Out[109]:
```
[0.04846649,
 0.042821117,
 0.03781131,
 0.0386842,
 0.055064,
 0.050130684,
 0.043984495,
 0.087888956,
 0.54818475,
 0.046964042,
 36,
 4]
```

Supervised Classification (Training Data Result)

- X = [train_vecs];
- Y = [predicted_labels];
- Result:

```
Logistic Regression Val f1: 0.869 +- 0.003
Logisitic Regression SGD Val f1: 0.855 +- 0.008
SVM Huber Val f1: 0.870 +- 0.003
```



Supervised Classification (Testing on Unseen Data

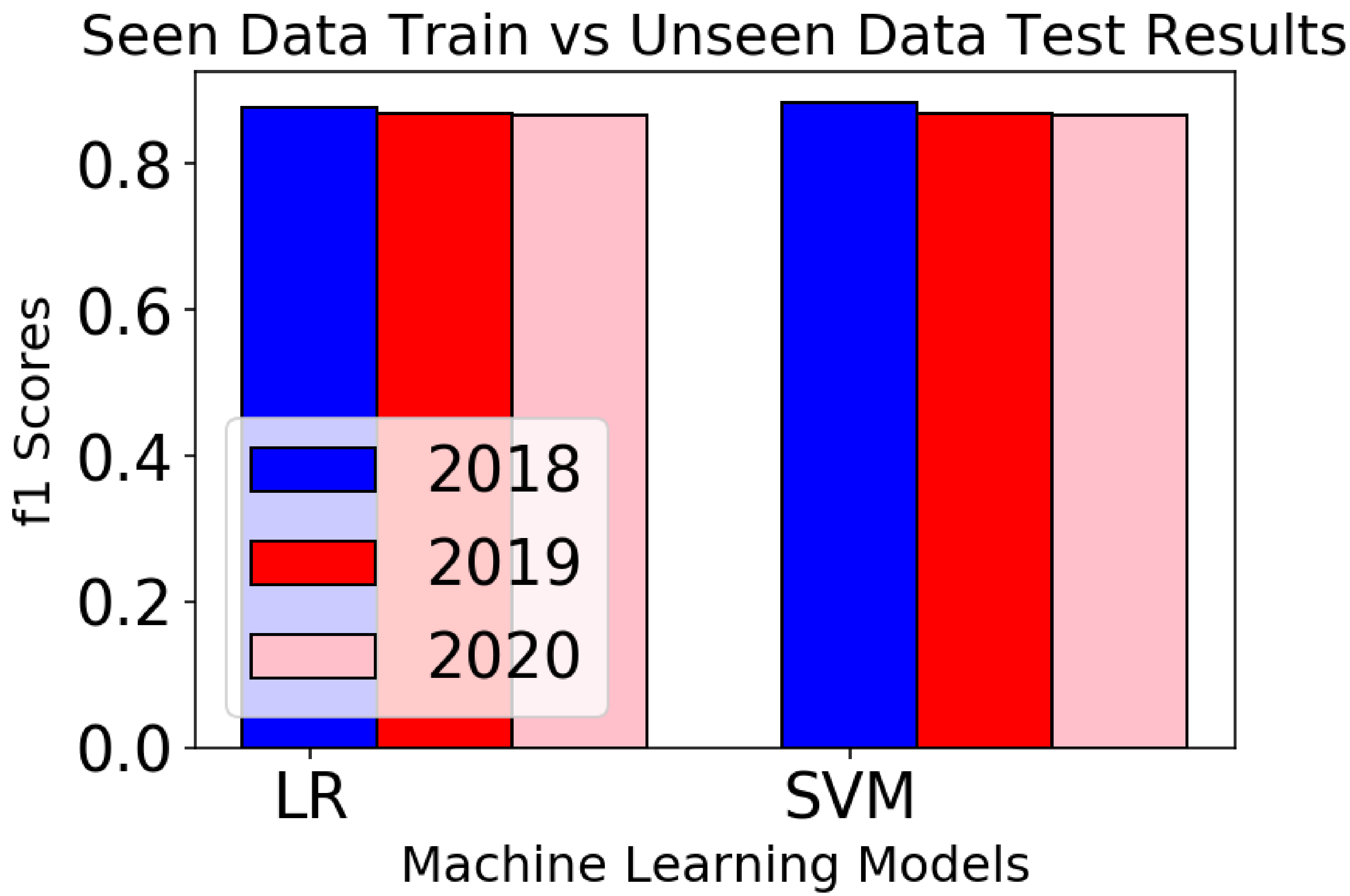
- For 2018:

```
0.8775611031997443
0.883026010151702
```

- For 2020:

```
0.8663699340718182
0.8665751454533569
```

Supervised Classification (On Test Data)



SHOCKING!!!!!!!

Conclusion

“ The analysis implicates the topic model for 2019 year can identify the la tent semantic structure that persists over time in this budget text domain”

Hypothesis Testing

- H0(null hypothesis) -> The ML models are similar and perform for all the year .
- H1 -> The ML models are truly different and perform differently.
- Condition for Hypothesis taken such that p-value threshold is p = 0.05

```
chi-squared: 10.861150070126227
p-value: 0.0009820269000594094
```

- Hence, the null hypothesis was rejected, as the models were completely different.

Next Word Recommender

- **Simulated text with markov chain method.**
- **A Markov chain is a simulated sequence of events. Each event in the sequence comes from a set of outcomes that depend on one another.**
- **For any sequence of non-independent events in the world, and where a limited number of outcomes can occur, conditional probabilities can be computed relating each outcome to one another.**
- **To generate a simulation based on a certain text, count up every word that is used. Then, for every word, store the words that are used next. This is the distribution of words in that text *conditional on* the preceding word.**

Next Word Recommender

https://drive.google.com/open?id=1J-O3GMuii8fL9DrOM0MvREFdU9eznQYk