Documents as Objects, Topic Modeling, Biclustering, and Ontology

Robin Singh

Northwestern University

Abstract:

This research uses the News Category Dataset v2 available for public use through Kaggle, and from the Random Forest Classification performed in the previous assignment, the TF-IDF had performed the best, and will hence be the "Reduced Matrix" for this assignment. For this assignment, we began by clustering our documents as objects with the K-Means algorithm, so that we may try to group similar articles together and evaluate the performance against the categories. Afterwards, we also conducted t-distributed stochastic neighbor embedding (t-SNE) for the multidimensional scaling, as well as hierarchical scaling to scale the documents and plot them to identify clusters and see how well they performed or were in line with the predetermined categories. We will also be conducting Topic Modeling on the Reduced Matrix, as well as "Biclustering" and creating an ontology to provide a "birds-eye view" of the world our corpus addresses.

Introduction:

This research was conducted to analyze the performance of K-Means scaling with t-SNE and hierarchical scaling to see how well the documents are clustered based on their similarities. These are complicated as many words are overlapping and similar to each other, so it may be difficult to get the clusters in-line with the predetermined categories. Still, this serves as a preliminary to more advanced methods to categorize documents, such as neural networks. Considering a business that is continually monitoring news and social media sources, gathering electronic documents relating to its products and services, as well as the products and services of its closest competitors there is a way to perhaps group these documents through scaling and clustering, however with these methods identified above, it still may not be the best way to get them in-line with preexisting categories, as the performance is just not there. More advance neural network models may be necessary to get more accurate results.

Literature Review:

Similar projects to this have been done by many people, such as an individual named Muneed, who created a blog about his project on DigitalVidya.com. He had created a text classification model, analyzing multiple techniques to classify the model, including Random Forest, Multinomial NB, and Logistic Regression. He had also clustered his data using K-Means clustering to see similar documents plotted together. After preprocessing all of the data and vectorizing his documents, he found the logistic Regression appeared to classify the best in his example.

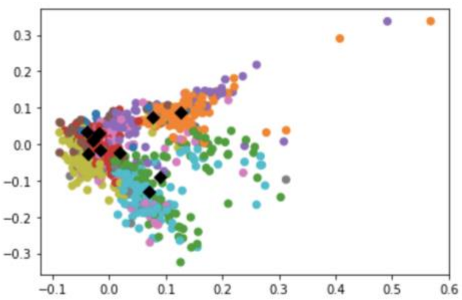
Methods:

The methods were defined in the assignment description, consisting of conducting K-Means clustering, t-SNE scaling and hierarchical scaling on the reduced to plot similar documents to see how well the clustering had worked. Afterwards, we were to conduct topic modelling on the reduced matrix, as well as “biclustering” to again to compare various ways of clustering our data to find similar data.

Results:

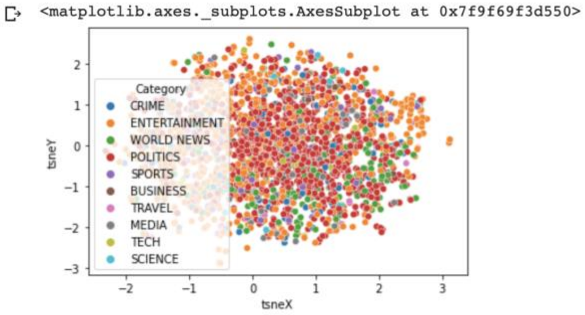
In order to easily plot our multi-dimensional data, I conducted Principle Component analysis so that it could provide coordinates to plot.

Our plot for the K-Means plot appeared as so:



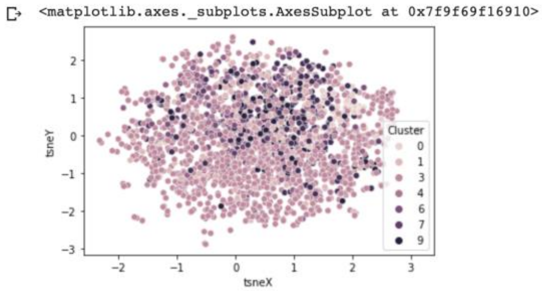
As you can see, there is a lot of overlapping occurring in our data. This may be because of the oversimplification the summarized descriptions of the news articles may have, allowing for not a lot of common terminology for the model to work with.

Our t-SNE plot appeared as so when plotted by Category:



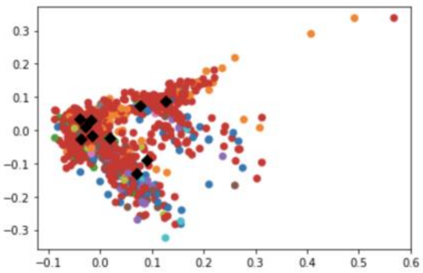
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f9f69f3d550>
```

It appeared as so when color-coded to the K-Means clusters:



```
<matplotlib.axes._subplots.AxesSubplot at 0x7f9f69f16910>
```

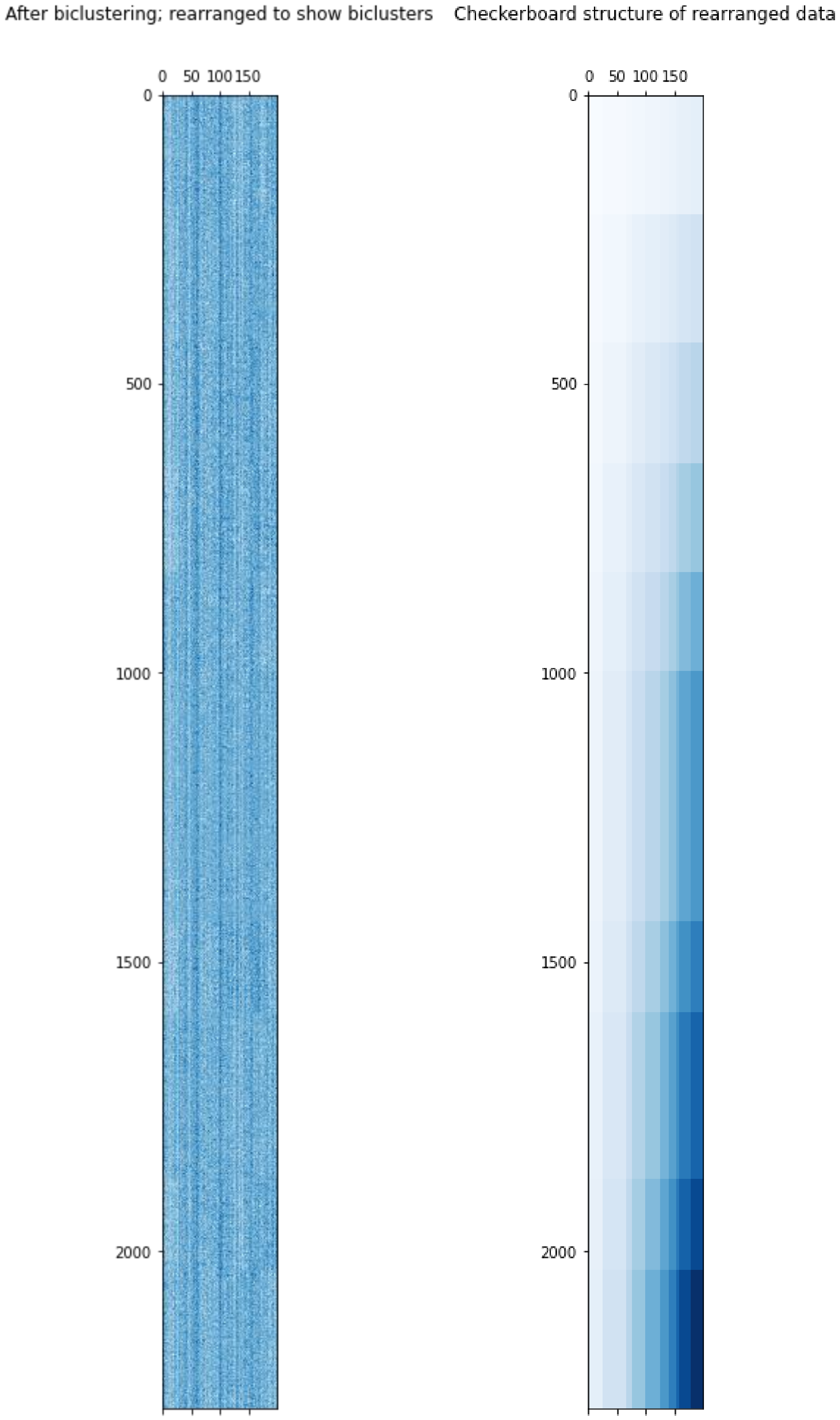
The hierarchical scaling plot appeared as so:



Our topic modelling consisted of these 10 topics, which I chose to be 10 to be on par with the 10 predetermined categories:

```
Topic: 0 Word: 0.068*"" + 0.027*"go" + 0.026*"want" + 0.024*"movi" +
0.024*"report" + 0.023*"peopl" + 0.022*"star" + 0.021*"also" + 0.018*"say"
+ 0.017*"student"
Topic: 1 Word: 0.035*"" + 0.029*"time" + 0.028*"show" + 0.027*"say" +
0.023*"talk" + 0.021*"need" + 0.020*"make" + 0.019*"alleg" + 0.018*"one" +
0.017*"public"
Topic: 2 Word: 0.034*"trump" + 0.034*"" + 0.026*"like" + 0.025*"hit" +
0.023*"michael" + 0.023*"said" + 0.020*"thing" + 0.020*"film" +
0.019*"meet" + 0.018*"man"
Topic: 3 Word: 0.035*"day" + 0.030*"america" + 0.025*"play" + 0.022*"love"
+ 0.022*"open" + 0.019*"one" + 0.019*"includ" + 0.019*"use" +
0.018*"victim" + 0.018*"stop"
Topic: 4 Word: 0.034*"year" + 0.034*"say" + 0.023*"last" + 0.023*"much" +
0.021*"got" + 0.020*"group" + 0.020*"month" + 0.019*"take" +
0.018*"democrat" + 0.016*"gun"
Topic: 5 Word: 0.067*"said" + 0.037*"hous" + 0.026*"white" +
0.025*"singer" + 0.024*"" + 0.020*"could" + 0.019*"he" + 0.019*"say" +
0.018*"next" + 0.018*"effort"
Topic: 6 Word: 0.034*"call" + 0.032*"get" + 0.026*"right" + 0.025*"fox" +
0.023*"question" + 0.021*"first" + 0.021*"good" + 0.020*"said" +
0.019*"happen" + 0.018*"news"
Topic: 7 Word: 0.035*"star" + 0.031*"school" + 0.031*"state" + 0.028*"new"
+ 0.026*"back" + 0.022*"lot" + 0.021*"first" + 0.021*"appear" +
0.021*"realli" + 0.020*"would"
Topic: 8 Word: 0.046*"presid" + 0.027*"one" + 0.023*"said" + 0.022*"two" +
0.021*"trump" + 0.021*"us" + 0.020*"reportedli" + 0.020*"critic" +
0.018*"" + 0.018*"senat"
Topic: 9 Word: 0.043*"trump" + 0.031*"said" + 0.030*"us" +
0.029*"comedian" + 0.027*"look" + 0.027*"know" + 0.021*"littl" +
0.021*"use" + 0.019*"would" + 0.016*""
```

And finally the results of the biclustering on the Doc2Vec data from assignment 2:

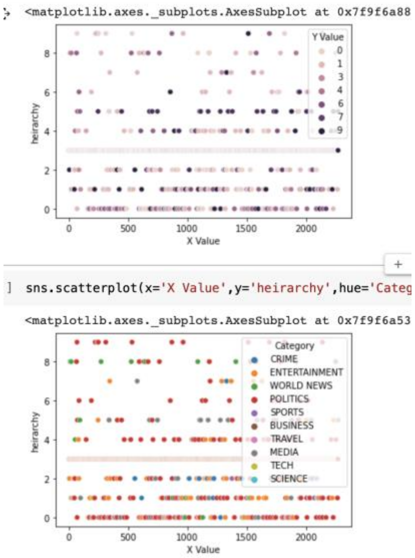


After biclustering; rearranged to show biclusters

Checkerboard structure of rearranged data

Conclusion:

From the results, it appears the models consist of a lot of overlapping terminology and material, getting the models confused on how to separate the topics. The K-Means plot by itself seems to work ok, providing some separation of topics, but still consists of a lot of overlapping. This is even prevalent in the t-SNE plot, however it seems to work better in terms of the clusters, as you can see the gradient getting darker as it reaches the upper-right corner of the plot, indicating some correlation with the K-means clustering data, but not with the predetermined categories. The hierarchical plot seems to spread out one topic all around, and seems to have bias, as seen in this plot here:



```
<matplotlib.axes._subplots.AxesSubplot at 0x7f9f6a88
```

```
] sns.scatterplot(x='X Value',y='heirarchy',hue='Categ
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f9f6a53
```

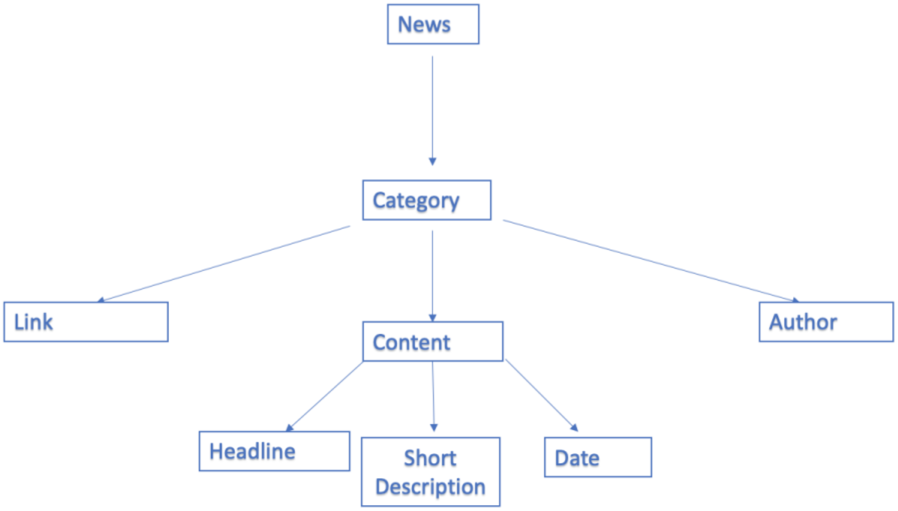
The political category is what the bias seems toward, and this could make sense considering the vast number of topics politics may cover, especially in the recent years. That is also prevalent in the topic modelling topics, which have the words “pres,” “trump,” and “America” in multiple topics. It seems easy for the clustering to get confused as such. The biclustering, however, did seem to show some clustering done well, as seen with the gradient like plot once rearranged.

This is because Doc2Vec works with a neural network, and thus had already assigned numbers to docs indicating how similar they were to each other. The more similar one document is to another, the more it’s values in its vector were closer, hence the gradient towards the bottom-right appearing in the rearranged data plot. Hence, I believe the neural network models are what will be most suited for this kind of analysis, which would be my focus for the next assignment.

Works Cited:

Muneeb. “Document Classification Using Python And Machine Learning.” Digital Vidya, 9 Jan. 2020, www.digitalvidya.com/blog/document-classification-python-machine-learning/.

Appendix:


News
Category
Link
Content
Author
Headline
Short Description
Date