Outputs: AI and Trump

Northwestern University

Ali Gowani

Output: AI and Trump

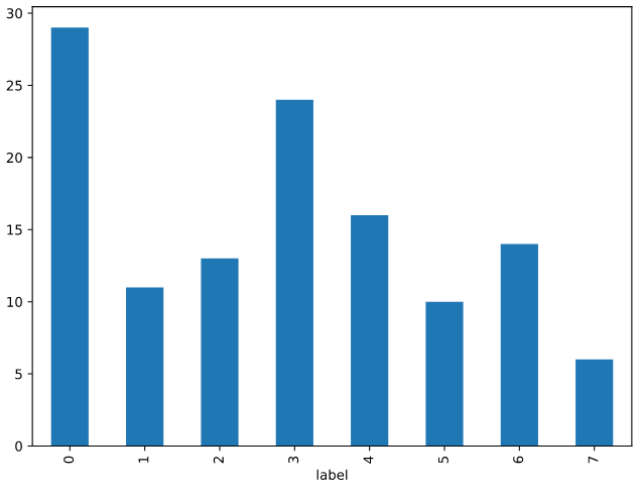
Introduction & Problem Statement

In our previous assignment, our focus was to determine which Reference Term Vector (RTV), would best suit our Data Source Items (DSI) as part of the overall corpus of the Trump Administration. We evaluated the results of Term Frequency – Inverse Document Frequency (TF-IDF) and Word2Vec. Our current assignment is to assess our clusters to determine whether they align to our documents, while keeping in mind that garbage values and noisy texts can incorrectly cluster our documents. My two documents were focused on Trump Administration and Artificial Intelligence (AI):

1. Document 1 (Doc1): New Trump Ruling Limits AI Surveillance Exports Over China Military Fears
2. Document 2 (Doc2): Trump administration's CES message: We're not interested in heavy AI regulation

After executing Paul's code with few modifications, the following bar plot represents the number of documents that were placed in the 8 clusters.

Figure 1:



My Doc1 was placed in Cluster 3 and Doc2 was placed in Cluster 6. We want to determine whether our documents clustered well. Did our documents cluster with other similar documents? Were there documents that were missing that were clustered elsewhere?

Summarize Algorithm and Key Features

We were provided with a corpus based on the submission by the students for topics relating to the Trump Administration. We loaded this into our model and conducted few functions to process the documents. These functions included, removing punctuations, filtering short tokens, turning words into lowercase and filtering stop words. Stop words, such as, “the”, “a”, “an”, “in,” were ignored when indexing. This allows the algorithm to drop common words that add little or no value when document matching (Brownlee, 2020). The algorithm identified top 10 terms for each cluster using K Means clustering based on TF-IDF matrix. The algorithm also used ngram range for TF-IDF vectorization. Ngram range is the number of words in a sequence. The ngram range that I used was between 1 and 3 (Sklearn, 2020). I felt that this was the most appropriate for my documents. The following figure shows the top 10 terms for Cluster 3 and Cluster 6 which were associated with our documents.

Figure 2:

Cluster 3, k=8, Doc1:	Cluster 6, k=8, Doc2:
trade	bulbs
china	administration
tariffs	environmental
chinese	schools
trump	climate
huawei	technology
manufacturing	standards
deficit	construction
agreement	projects
good	energy

Review and Interpret Results

When I compare what the program shows as the top terms for the clusters compared to the top terms (TF-IDFs) for my documents (Figure 3), I do not feel that our documents clustered well. For example, it seems that my Doc1 was clustered with other documents relating to China more so than anything else, including Artificial Intelligence. My Doc2 seemed to be clustered with documents that were focused on bulbs and environment.

Figure 3:

	AAG_Doc1_Trump Limits AI Exports To China.docx
surveillance	0.2884
technology	0.1943
china	0.1314
restrictions	0.1178
industry	0.1144
export	0.0921
giants	0.0865
technologies	0.0730
companies	0.0698
exports	0.0686

	AAG_Doc2_Trump administration not interested in AI regulation.docx
selfdriving	0.2427
technology	0.1344
artificial	0.1294
artificial intelligence	0.1294
companies	0.1208
ensure	0.1100
autonomous	0.0988
autonomous vehicle	0.0988
intelligence	0.0979
vehicle	0.0910

It is interesting to note that my Doc2, which was in Cluster 6, also had AI related documents, such as, trump-administration-plan-for-ai.docx. Perhaps, it is not that our documents did not cluster well but rather there were not enough documents to have an AI related cluster.

This got me thinking whether adjusting the number of clusters in our K Means algorithm would make a positive impact to better clustering for our documents. First, I increased the number of clusters from 8 to 12. My rational was that this would allow the algorithm to hone-in to the subject of AI.

Figure 4:

Cluster 3, k=12, Doc1:	Cluster 9, k=12, Doc2:
china	court
trade	providers
tariffs	judge
chinese	internet
trump	schools
huawei	nominees
agreement	rules
talks	principles
farmers	broadband
companies	appeals

I was certain that my documents would be better clustered and at the least, clustered together. Neither of those occurred when I increased the number of clusters to 12 (k=12). In addition, some of the terms in these clustered were completely not related, such as, farmers and schools.

My next approach was to decrease the number of clusters from 8 to 6. My rational in this scenario was that if the clusters were wide enough, then perhaps it would at the least capture both of my documents relating to AI.

Figure 5:

Cluster 0, k=6, Doc1:	Cluster 4, k=6, Doc2:
trade	children
china	bulb
tariffs	percent
chinese	technology
trump	environment
huawei	schools
manufacturing	standards
deficits	climate
goods	federal
economy	administration

It was surprising to see that even when decreasing the number of clusters, my documents did not end up in the same cluster. Also, the terms in these clusters, seem to be at a much higher level, where we see more general terms, such as, percent, administration, economy, etc.

Summarize Insights and Findings

This was an interesting assignment to try different things in order to see whether we can nudge our documents in the appropriate clusters or see where our documents align relative to similar documents in the corpus. The more analysis I conduct, the more I am realizing that my documents are the root of the problem. While the topic of AI is captured in both of my documents, it may not be the central concept. In addition, while the area of AI was discussed in each document, it may have been a subset for the wider topic of the document, from exports to China or regulations when it comes to technology.

References

sklearn.feature_extraction.text.TfidfVectorizer¶. (n.d.). Retrieved February 16, 2020, from https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.TfidfVectorizer.html

Brownlee, J. (2019, August 7). How to Prepare Text Data for Machine Learning with scikit-learn. Retrieved January 30, 2020, from https://machinelearningmastery.com/prepare-text-data-machine-learning-scikit-learn/