Impeach-ary – Using Extractive Methods to Summarize Political Text

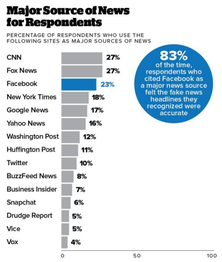
Arjun Karanam, Michael Elabd, Ronak Malde
CS 221 Final Project

Problem / Motivation

Problem
-Due to the large amount of information available for consumption, media has turned to sensationalized headlines to quickly pass information to people.

-Lack of civic awareness/engagement because it is far too time consuming to watch full videos of senate proceedings, or for our area of interest, impeachment hearings.

Project Goal: Create an algorithm that summarizes each witnesses' statement during the Trump impeachment hearings using unsupervised learning



Pre-Processing

Input
Full Articles

Pre-Processing
- Split into sentence
- Lowercase all words
- Remove stopwords

Output
TF-iDF or Vector

Rationale: Words such as "great" and "good" maybe structurally different but their meanings are equivalent. Representing sentences on a higher dimensional space can serve to give us an understanding of what each sentence means and how close that meaning is to other sentences.

Data

Data Collection: For our project, we required two main datasets:
1)A generic dataset of political articles
2)Impeachment hearing transcripts

We acquired the first through combining a Kaggle news article dataset with hand-picked political articles from Reuters and CNN datasets.

The impeachment hearing transcripts were scraped manually.

Baseline

Algorithm: Rather than using any heuristics, simply pick every n sentence.

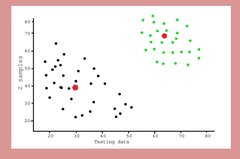
*"**The next round of public impeachment hearings is scheduled for Friday, with former US Ambassador to Ukraine Marie Yovanovitch preparing to take center stage**. She is a career diplomat who was abruptly pulled from Kiev last spring after a personal order from President Donald Trump. He made the decision after a months-long public campaign against Yovanovitch, led by his attorney Rudy Giuliani and others in the right-wing media. **Yovanovitch testified behind closed doors last month, but Friday's public hearing will be different."***

Algorithms and Approaches

K-Means

Method:

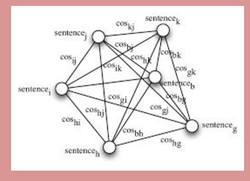
- Make a Doc2Vec model trained on all words and transform sentences to vectors
- Run the K-means algorithm
- Run PCA to decrease the dimensionality
- Output: Sentence closest to each centroid in chronological order



Weighted Heuristic

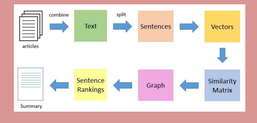
Method:

- Tokenize each sentence into words and calculate their weighted frequency
- Use weighted frequencies to calculate a "sentence score"
- Output: All sentences above a threshold are placed in summary



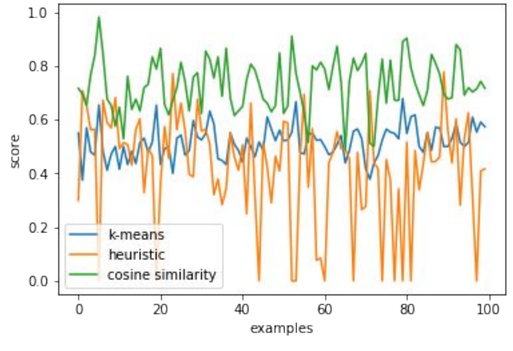
Cosine Similarity

Method:
- Find a word embedding in the form of a vector for each sentence
- Create a Similarity Matrix
- Convert matrix into a graph (Vertices – sentences, Edges – Similarity Scores)
- Run the PageRank algorithm on graph
- Output: Top N sentences

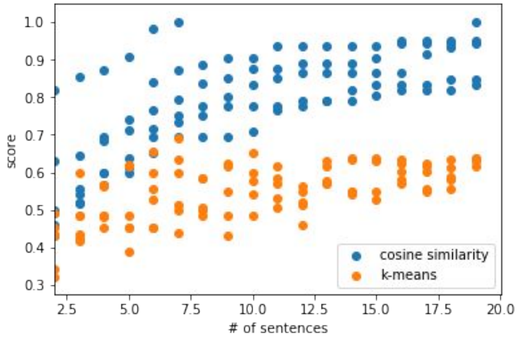


Results

Models over iterations vs. Unit Overlap Score



Number of Summarizing Sentences vs. Score



Analysis

Unit Overlap: Computes the most common terms and divides it by the total term count to get a correlation factor

Lexical Semantic Analysis: Analyze the distance between two strings (article and generated summary) using an embedding model and a similarity matrix

Results:
-Overall, we saw that the Cosine Similarity model performed the best on the basis of Unit Overlap evaluation metric
-However, the performance of the model was very unstable from iteration to iteration
-The heuristic performed the worse, probably because it's prone to redundancy

Future

Create a Website – Public facing app to update the public on the latest impeachment summaries

Retrain on Reference Summaries – The lack of which lead to us not being able to use industry standard evaluation metrics

References

Barzilay, R.—Elhadad, M.: Using Lexical Chains for Text Summarization. InProceedings of the ACL/EACL '97 Workshop on Intelligent Scalable Text Summa-rization, Madrid, Spain, 1997, pp. 10–17

Mihalcea, R.—Tarau, P.: Text-Rank – Bringing Order Into Texts. In Proceedingof the Comference on Empirical Methods in Natural Language Processing, Barcelona,Spain, 2004..