Topic Analysis on Obama Briefings

Overview

The goal of the project is to discover the abstract topics that occur in a collection of Obama briefings to check which topic Obama mentioned the most.

Client

News company : who want to know which area Obama considered important and mentioned often in his briefings.

Social media like facebook, youtube, Instagram and ect: Topic analysis can be used to analyze which topic people most interested at that time.

Data

Obama's briefings were found in the whitehouse webpage. We use the data file at github, which is the download from whitehouse homepage, " https://www.whitehouse.gov/briefings-statements/ ". The data file can be found at " https://github.com/mahmoud/briefings "

Briefings.jsonl : 1778 Obama's briefings from 2009 to 2016.

Data Cleaning

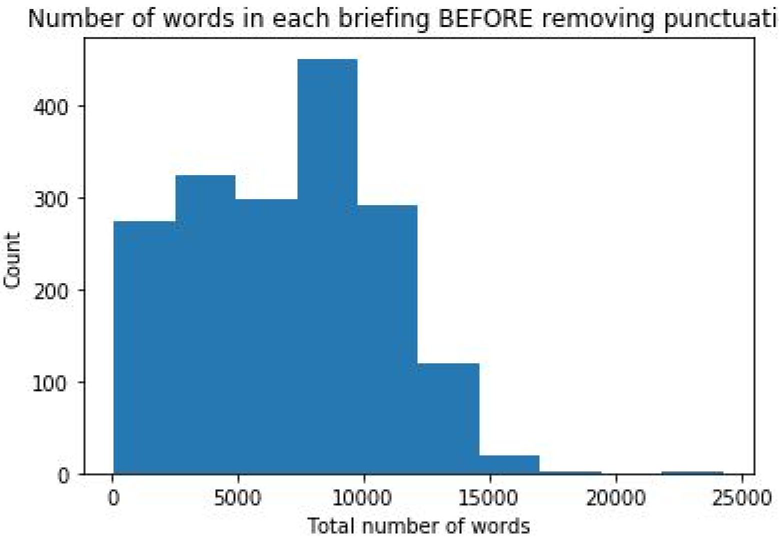
- Punctuation : punctuation such as "a", "!" , "?", ets were removed at each contents of briefings using "string.punctuation" provided from Python. After removing punctuation, 54.3% was reserved.
- Unicode implementation: To have words encoded using Unicode encoding, I encoded text type to utf-8 and remove any numbers and also change every words to lower cases.

Data Preparation

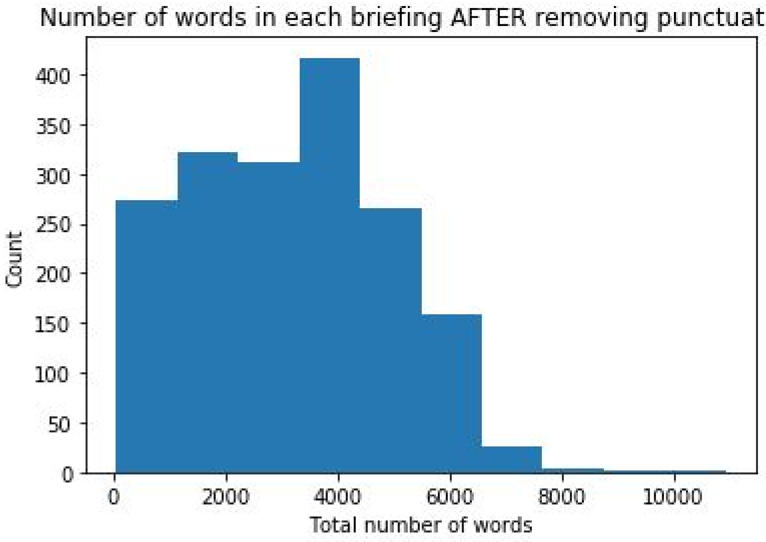
- Tokenization : Tokenized every words in contents
- Removing morphological affixes: "stemmer.stem(token)" was used to remove any unnecessary morphological affixes for better analysis
- Construct word vs ID mapping table to map same tokens to the same IDs for managing efficiently

Data Storytelling

Let’s check the number of words before removing punctuation. The mean is 7016.93 and the minimum and maximum are 101 and 24271. The frequency bar plot is like below.



Let’s check the number of words after removing punctuation. The mean is 3207.11 and the minimim and maximum are 62 and 10929.



From two bar plots above, we can see that graphs look alike. After removing punctuation, about 50 % was removed.

Machine Learning

First, mapping table on "descriptions" vs ID was created to efficiently apply values to models. Then,dictionary was converted into a bag-of-words. The result, corpus , is a list of vectors equal to the number of documents. In each document vector is a series of tuples

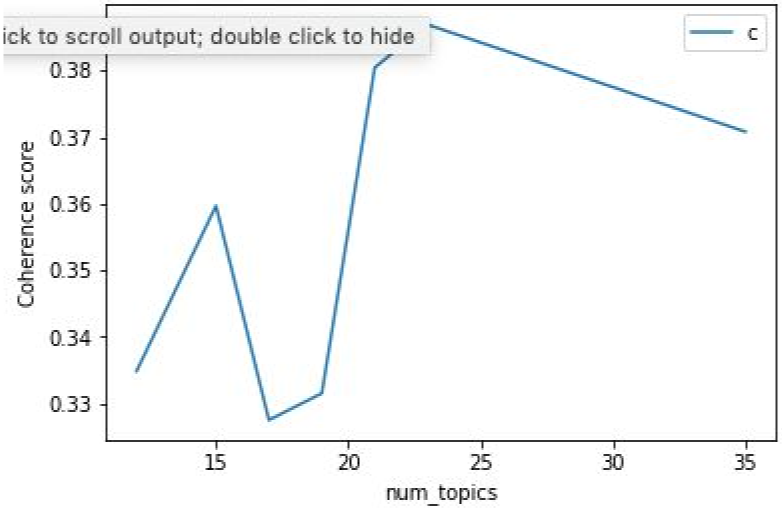
To run any mathematical model on text corpus, it is converted into a matrix representation.

LDA (Latent Dirichlet Allocation)

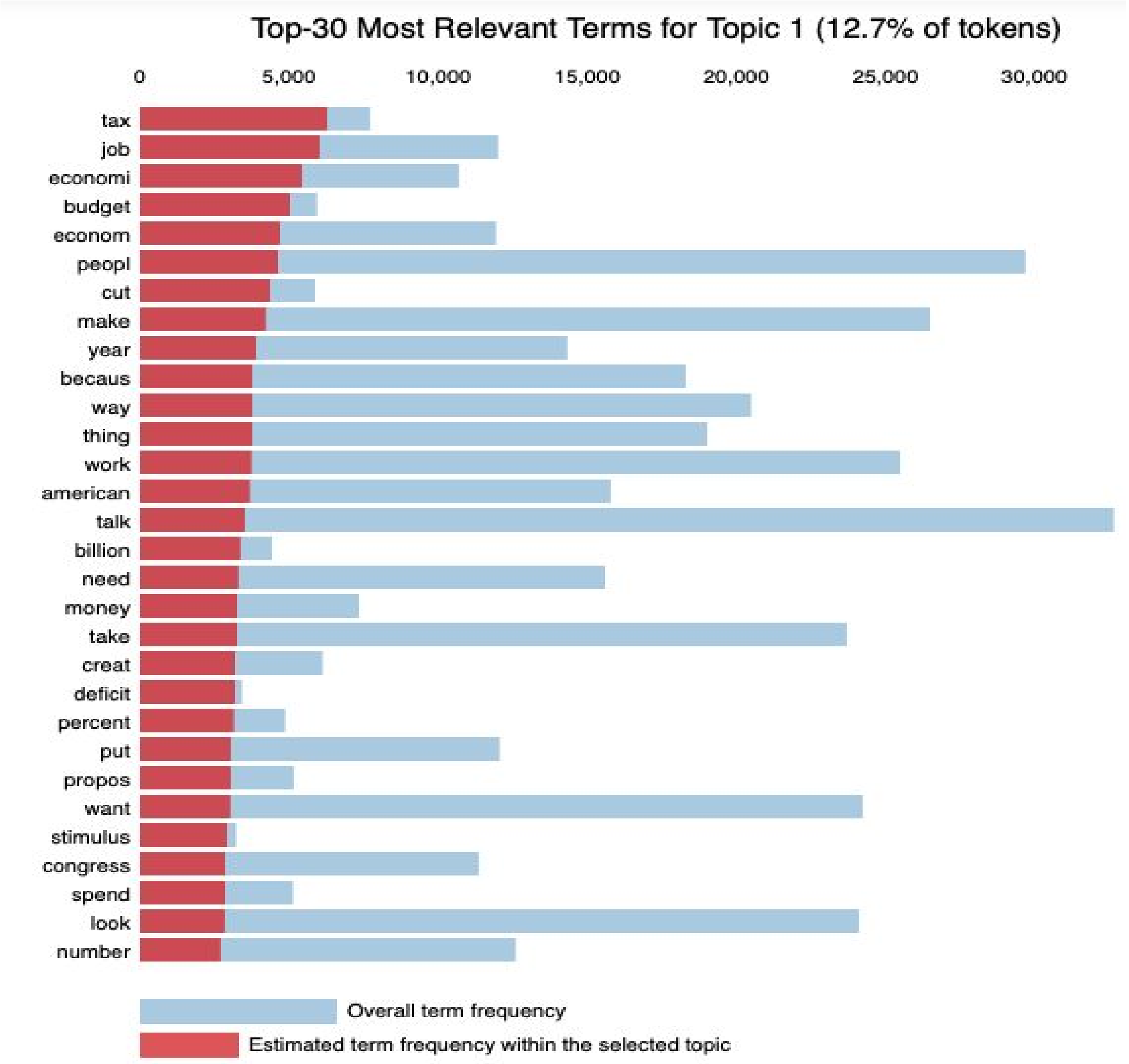
This is one the most popular topic modeling algorithms today. It is a generative model in that it assumes each document is a mixture of topics and in turn, each topic is a mixture of words.

LDA require a num_topics parameter (set to 200 by default) to determine the number of latent dimensions after the SVD. So several numbers of topics were tested on the model to pick the best one. Coherence socre was used to measure the model. Topic coherence in essence measures the human interpretability of a topic model. Traditionally perplexity has been used to evaluate topic models however this does not correlate with human annotations at times. Topic coherence is another way to evaluate topic models with a much higher guarantee on human interpretability. Thus this can be used to compare different topic models.

The number of topics which are from 12 to 35 were applied to the model to see the best model. From the graph below, the model with 23 numbers of topics are the best model and the coherence score is 0.39.



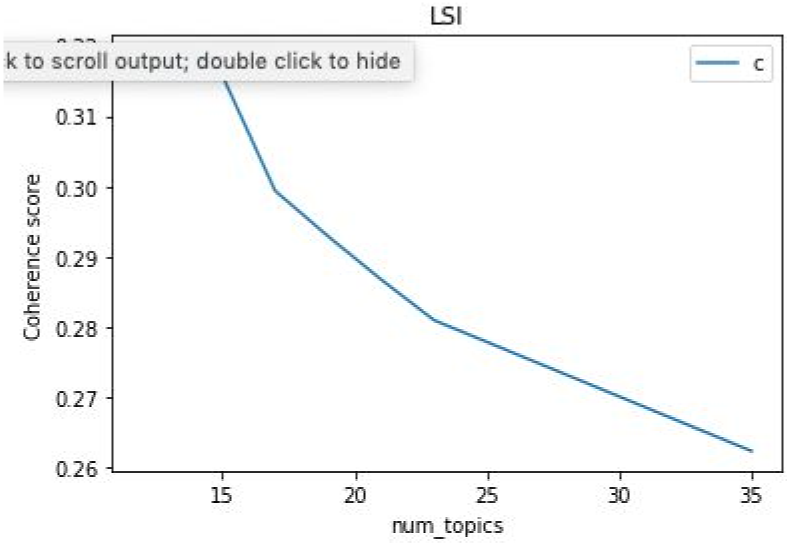
The data visualization is like below.



LSI (Latent Semantic Indexing)

This is a useful topic modeling algorithm in that it can rank topics by itself. Thus it outputs topics in a ranked order. The difference between LDA and LSI is that LSI learns latent topics by performing a matrix decomposition (SVD) on the term-document matrix. LDA is a generative probabilistic model, that assumes a Dirichlet prior over the latent topics. In practice, LSI is much faster to train than LDA, but has lower accuracy.

The model also needs to decide the number topics to apply to the model. From the graph, the number of topics of 13 has the optimal result. The coherence score is 0.32.



Conclusion