Dwipam Katariya, Sanjana Agarwal and Krish Mahajan

ASSIGNMENT 6:

a) We used Gensim api in python for topic modeling.

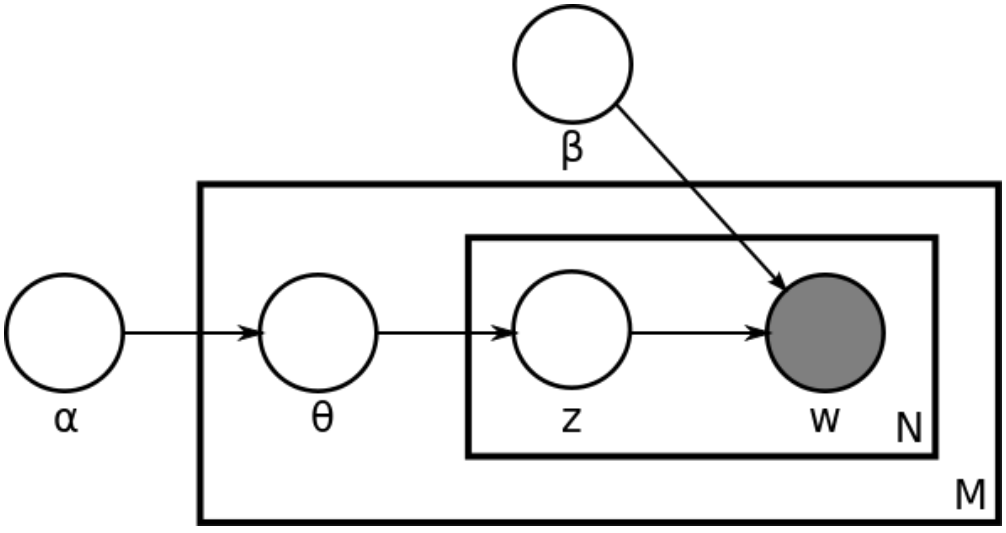
b) The questions we thought of investigating are:

- How are the topics distributed for Election Candidates conversation over the debate?
- After generating the topic distribution how accurate are we?
- Do the topics make sense for the given Candidate conversations?

Data Source : https://www.kaggle.com/mrisdal/2016-us-presidential-debates
Data Description : Data is a .csv file that has the the conversation transcript for each Election candidate.

c)
We used LDA algorithm for Topic modeling.
In natural language processing, **Latent Dirichlet allocation** (**LDA**) is a generative statistical model that allows sets of observations to be explained by unobserved groups that explain why some parts of the data are similar. For example, if observations are words collected into documents, it posits that each document is a mixture of a small number of topics and that each word's creation is attributable to one of the document's topics.
Reference: https://en.wikipedia.org/wiki/Latent_Dirichlet_allocation,
https://www.quora.com/What-is-a-good-explanation-of-Latent-Dirichlet-Allocation

Plate Notation representing the LDA model



Thus, it represents documents as mixtures of topics that consist of certain words with certain probabilities.
It assumes that the documents are created in the following way:

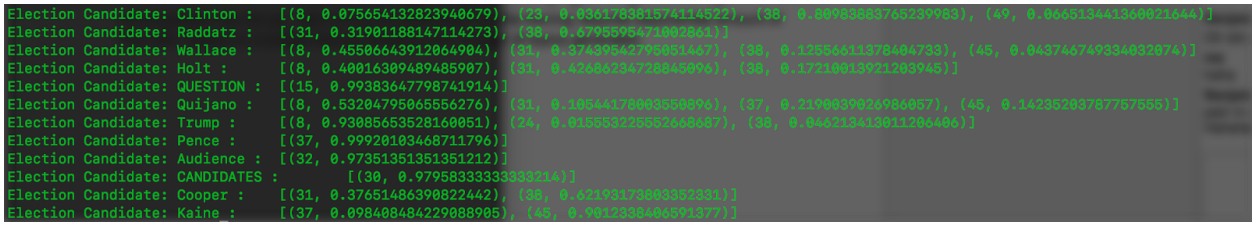
- Decide on the number of words that the document will have (by Poisson Distribution)
- Choose a topic mixture for the document

- Generate each word in the document by: first picking the topic and then using the topic to generate the word itself.

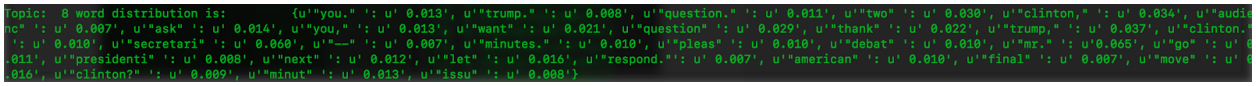
LDA then tries to backtrack from the documents to find a set of topics that are likely to have generated the collection.

After generating the topics using gensim package in python for each election candidate, some topics generated did make sense, however topics with high words will **mr., go, think, just, want** had lower interpretation of the allocated topic. While word distribution for the topics such as **Country, Tax, Pay, American, People, Enforce, Terrorist** did make sense. One thing we denoted was, Clinton has higher probability for topic that has higher probability of word Trump, and V.i.Z. We generated 50 topics with 30 words that describes the topic. Hence, it does help us to understand the emotions, personality of each election candidate. If we were not computer savy, using the api gensim functionality is difficult as it requires knowledge of object oriented programming with the way LDA is encompassed into the api. Also tuning the parameters for the training requires exponential amount of time with respect to increase training iterations and number of topics generated.

We have multinomial distribution over all the assigned documents for a respective document(candidate)



We also generate words per document with Topic ID as follows:



Please run the program as follows placing debate.csv in the same folder as topic_model.py

```
python topic_model.py debate.csv
```