Deliverable 3

Fine-tuning LDA model:

I have modified the preprocessing step. Using gensim's built-in Phrases/Phraser model, I was able to add bigrams and trigrams to better enhance my results. It looks through the corpus and automatically creates n-grams based on the frequency of the co-occurrence of two or more words in the corpus. For instance, my model can now distinguish between "machine learning", "machine", and "learning" by grouping "machine learning" into "machine_learning" and treating it as one single word when they occur together consecutively a sufficient number of times in the document (chosen to be 20, can be tuned as it is a hyperparameter).

Here is a snippet of the text after adding n-grams.

Another modification I did was increase the number of "epochs" of the LDA model. This allows it to "pass" over the training corpus several times before deciding on the final weights/probabilities it will assign to each word in each topic and to each topic of each document. In general, we want to increase the number of epochs to be above the default of 1 but ideally stop at a number of passes that will not lead the model to overfit the training set. In my previous deliverable, I noticed that many topics had the same words assigned to them, therefore there was no clear distinction between topics (similar to underfitting). We can reduce this word overlap between different topics by increasing the number of passes (epochs) that the model makes during training.

Benefits of increasing number of epochs:

1. Improved convergence: By running more epochs, the model has more opportunities to explore and eventually converge on a stable set of topic assignments and word probabilities. This can lead to more consistent and reliable results.
2. Better capture of topic nuances: As the model continues to update the topic assignments and word probabilities over multiple epochs, it can potentially capture more subtle nuances in the text data that may have been missed with fewer epochs.

Evaluation Metric:

Perplexity as an evaluation metric is not deemed "good" since it does not always align with human interpretation of the validity of the topics assigned to the documents (which is what we are interested in). Perplexity captures how surprised a model is by new data it has not seen before and is measured as the normalized log-likelihood of a held-out test set. More intuitively, perplexity asks how well does the model represent or reproduce the statistics of the held-out data? However, recent studies have shown that predictive likelihood (perplexity) and human judgment are often not correlated, and even sometimes anti-correlated. This limitation of perplexity led me to choose a different metric to evaluate my model: topic coherence. There are many different variations of coherence metrics that use different formulas, but the underlying concept is more or less the same between all of them. Coherence is a number between 0 and 1. It can be measured within a single topic or across all the topics "discovered" by the model. The coherence of a single topic is calculated using a measure of semantic similarity between the top words in the topic. One common measure of coherence is the **Pointwise Mutual Information** (PMI) score, which calculates the co-occurrence frequency of pairs of words in a corpus and compares it to their expected co-occurrence frequency by chance. It compares the probability of two words occurring together in the same document to what this probability would be if the occurrences were independent probabilities. The PMI score can be used to evaluate the coherence of a single topic or the average coherence of all topics in the model.

I used gensim's 'c_v' score (other methods to calculate coherence could have been chosen). I obtained a coherence score of 0.367 for the training set. According to the literature, there is no "good" or "bad" coherence score. What matters is that we maximize it. From what I have researched though, a score of 0.35-0.4 is what is usually obtained by these topic models and anything above 0.4 is considered "good". I still do not fully have a grasp of the exact math behind coherence but will work on understanding that before the fair.

In regards to the TF-IDF/kmeans model, I did not have time to fix it. Also, I found a more interesting model called BERTopic which should supposedly be easy to implement. It utilizes TF-IDF preprocessing too which I have already done. It is also said to be a better model than LDA. This is why I thought it would be more interesting to do a side-by-side comparison between LDA and BERTopic rather than kmeans which is known to be bad for topic modeling generally. I should be able to have it done before the fair.

Final Presentation: Since my project does not require input from the user, I found it more convenient to prepare a slideshow instead of a webapp to explain the methodology and results of my model.