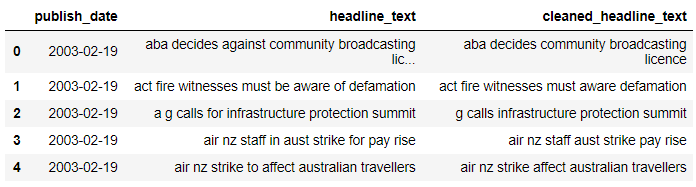
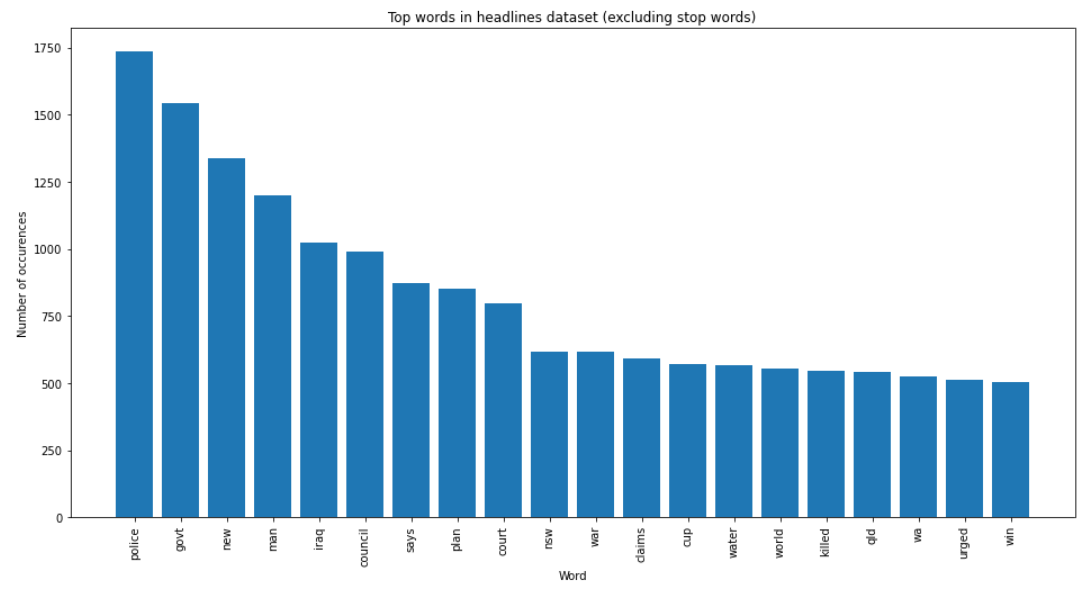
1. Data Preprocessing:
   1. Import all the necessary libraries.
   2. Load the datafile ‘abcnews-date-text.csv’, which was downloaded from the Kaggle. Due to the computational issue, downsizing the dataset into the first 50,000 records is a suitable solution.
   3. Remove the stop words for all the records, which I reference the stop words list from <https://gist.github.com/sebleier/554280> and <https://stackoverflow.com/a/47091490/4084039>.
   4. Create a new feature for the cleaned headline text, name as ‘cleaned\_headline\_text’. The first updated 5 records are as follow:



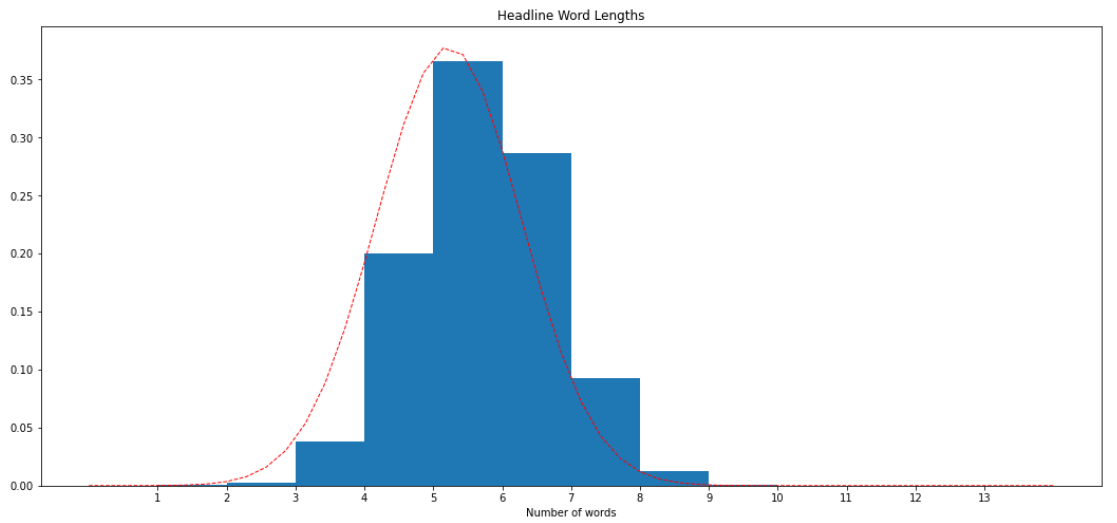
1. First Visualization:
   1. Plot the top 20 words that appear the most from the records. We observe that the word ‘police’ appears the most frequent for such a dataset:



* 1. Calculate the total number of words and the mean number of words per headline:



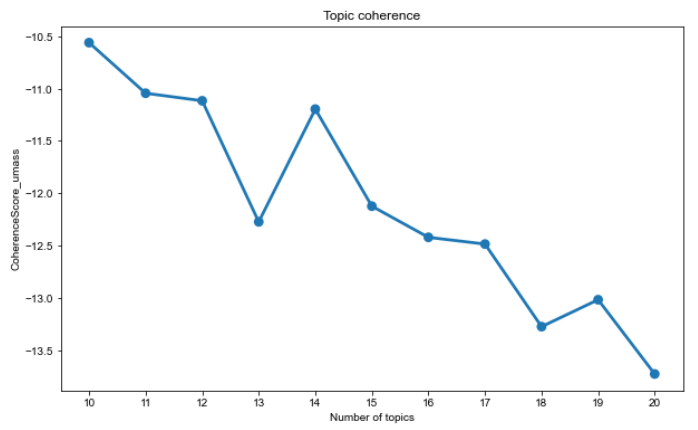
* 1. Plot the frequency of the headline word lengths distribution by using a standard deviation. We observe that 5 words have the most frequencies among all the headlines:



1. Topic Modelling Using Gensim Library:
   1. Create a term dictionary of the corpus, in which every unique term is being assigned to an index.
   2. Covert a list of headlines (corpus) into a headline term matrix using the prepared dictionary.
   3. For the coherence value of all the algorithms that will be mentioned, we use u\_mass as the indicator for the algorithm performance in this dataset. The reason being is that u\_mass generally compute in a shorter time than using the c\_v indicator. Hence, due to the computational limitation, u\_mass would still give us an accurate result.
2. LDA (Latent Dirichlet Allocation) Algorithm:
   1. Build the LDA model by initializing the parameters such as corpus, id2word, and num\_topics. For the initialization stage, we set the num\_topics to 10.
   2. Compute the coherence score for the LDA model using ‘u\_mass’:



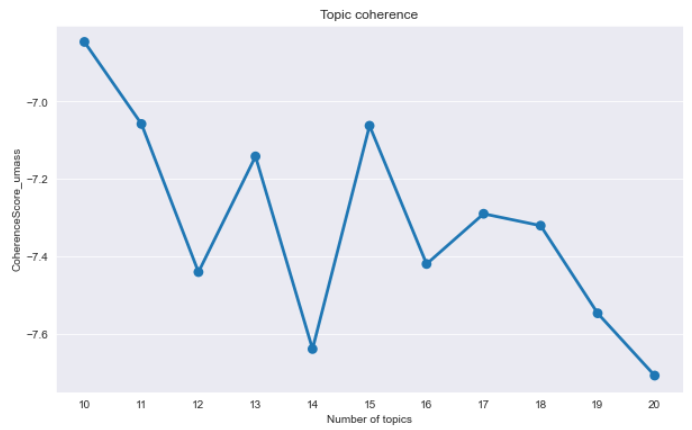
* 1. For a more accurate parameter value for num\_topics with respect to its coherence value, plot the coherence value for the num\_topics of 10 up to 20. The larger a coherence value, the better performance for the corresponding number of topics. For this algorithm, we observe that 10 topics perform the best, as the coherence value is the highest:



1. Latent Semantic Indexing (LSI) Algorithm:
   1. Build the LSI model by initializing the parameters such as corpus, id2word, and num\_topics. For the initialization stage, we set the num\_topics to 10.
   2. Compute the coherence score for the LSI model using ‘u\_mass’:



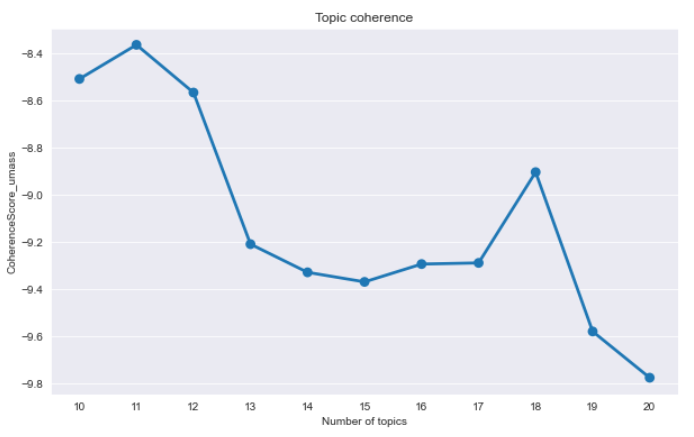
* 1. For a more accurate parameter value for num\_topics with respect to its coherence value, plot the coherence value for the num\_topics of 10 up to 20. The larger a coherence value, the better performance for the corresponding number of topics. For this algorithm, we observe that 10 topics perform the best, as the coherence value is the highest:



1. Non-Negative Matrix Factorization (NMF) Algorithm:
   1. Build the NMF model by initializing the parameters such as corpus, id2word, and num\_topics. For the initialization stage, we set the num\_topics to 10.
   2. Compute the coherence score for the NMF model using ‘u\_mass’:



* 1. For a more accurate parameter value for num\_topics with respect to its coherence value, plot the coherence value for the num\_topics of 10 up to 20. The larger a coherence value, the better performance for the corresponding number of topics. For this algorithm, we observe that 11 topics perform the best, as the coherence value is the highest:



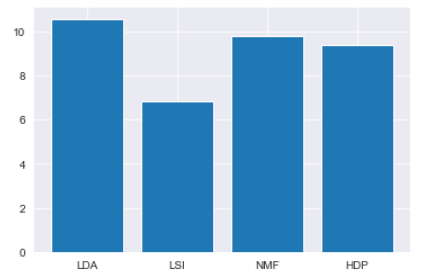
* 1. Compute the coherence score for the NMF model using ‘u\_mass’ with 11 topics:



1. Hierarchical Dirichlet Process (HDP) Algorithm:
   1. For this algorithm, it does not depend on the number of topics. Thus, we can directly compute the coherence score for the HDP model using ‘u\_mass’:



1. Evaluating All Four Models Performance:
   1. Extract the maximum coherence value from LDA, LSI, and NMF algorithm’s corresponding lists. Convert all the coherence values of all four algorithms into an absolute value (from negative to a positive value) for plotting purposes. From the resulting plot, we observe that LSI performs the best for this dataset, corresponds to the largest coherence value. Hence, we conclude that LSI is the optimal algorithm:



1. Visualize the LSI Model:
   1. To conclude our observation, we plot the top words from each topic, resulted from the LSI model. We observe that the word ‘police’ and ‘council’ appear the most from the given dataset of headlines:

