Framework for NLP:

|  |  |  |
| --- | --- | --- |
| Steps | Project | Group decision |
| **Raw Text:** Start with the raw data. | Only title or full article |  |
| **Tokenization:** Separate the words from paragraphs, to sentences or to individual words. | PySpark Machine Learning (ML) library.  From pyspark.ml.feature import Tokenizer  BERT, GPT-2, XLNe – BERT has been suggested in different articles. |  |
| **Stop Words Filtering:** Remove common words like "a" and "the" that add no real value to what we are looking to analyze. | StopWordRemover() in PySpark |  |
| **Term Frequency-Inverse Document Frequency (TF-IDF):** Statistically rank the words by importance compared to the rest of the words in the text. This is also when the words are converted from text to numbers. | The first is by CountVectorizer, which indexes the words across all the documents and returns a vector of word counts corresponding to the indexes. The indexes are assigned in descending order of frequency. For example, the word with the highest frequency across all documents will be given an index of 0, and the word with the lowest frequency will have an index equal to the number of words in the text.  The second method is HashingTF, which converts words to numeric IDs. The same words are assigned the same IDs and then mapped to an index and counted, and a vector is returned. For our Python example, if it gets a numerical ID of 4278 and it appeared 20 times, the vector would be 4278 : 20. |  |
| Machine Learning | Naïve Bayes Classifier  SVM  Neural network |  |

Have learned it in class