QUESTION 1: To get started, plot a histogram of the number of training documents for each of the 20 categories to check if they are evenly distributed.

A close up of a logo

Description automatically generated

The histogram for the 20-news category in the dataset. The number of training documents per category are roughly the same except for a few. Those that are do not have roughly the same number of documents are in the politics and religion section.

QUESTION 2: Use the following specs to extract features from the textual data:

• Use the “english” stopwords of the CountVectorizer

• Exclude terms that are numbers (e.g. “123”, “-45”, “6.7” etc.)

• Perform lemmatization with nltk.wordnet.WordNetLemmatizer and pos tag

• Use min df=3

Report the shape of the TF-IDF matrices of the train and test subsets respectively

After lemmatization, the TF-IDF matrices of train and test subsets are (4732,18488) and (3150,18488) respectively. Even after lemmatization, some still made it through the filter because of strange patterns. For example, chain \* are counted as word even though it shouldn’t be. This could be because of the default behavior of the lemmatizers. If it can’t be found in the dictionary it still includes it as a word.

QUESTION 3: Reduce the dimensionality of the data using the methods above

• Apply LSI to the TF-IDF matrix corresponding to the 8 categories with k = 50; so each document is mapped to a 50-dimensional vector.

• Also reduce dimensionality through NMF (k = 50) and compare with LSI:

Which one is larger, the error in NMF or LSI? Why is the case?

The sum of square error of LSI is 3328.02 and error for NMF is 3358.44 NMF has a higher error than LSI because NMF only outputs positive coefficients so it is possible to lose information due to this constraint.