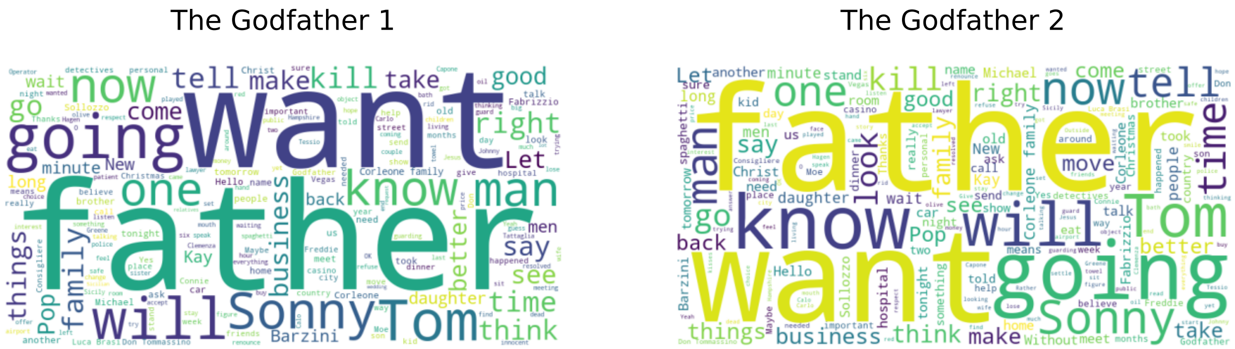
Task 2

WordCloud is a visual representation of text data where words are displayed in varying sizes based on their frequency in the given text. The more frequent a word is, the larger it appears in the cloud.



The Godfather 1

The Godfather 2

Figure 1: WordClouds for Godfather movie scripts

Plot analysis

The similarity in the most frequent words in Figure 1, such as ”father,” ”want,” ”will,” ”going,” ”know,” ”Tom,” and ”Sonny,” shows that I have been successful in scraping the lines of character Michael in Godfather movie scripts from two different sources. The differences in less frequent words between the two WordClouds reflect the difficulties I encountered while scraping the lines from the second source.

The problem was that the narrator’s lines weren’t separated from the lines of the characters. Even I as a human had a hard time determining if a certain line belonged to the character or the narrator. The break between the two was purely context-based. I was able to observe that when the narrator’s lines included the name of a character, the name was all in capital letters. I used this information to eliminate at least some parts of the narrator’s lines and achieved better results.

Task 3

Describe your understanding of WordPiece tokenization.

WordPiece tokenization is a subword tokenization technique that is similar to BPE. It differs in the way the score for each candidate token is calculated. WordPiece calculates the score by using the following formula:

$$score = \frac{freq\_of\_pair}{freq\_of\_first\_element \cdot freq\_of\_second\_element}$$

First, WordPiece adds special tokens used by the model to the vocabulary followed by all characters that occur in the corpus. The initial alphabet thus contains all the characters present at the beginning of a word and the characters present inside a word preceded by the WordPiece prefix (##). Next, it calculates the score for all pairs of elements and adds the element with the highest score to the vocabulary. This process is then repeated until the desired vocabulary size is reached. Even though WordPiece calculates scores for only pairs of elements (not triplets or more), thanks to its repetitive nature and adding the created elements to the vocabulary, it creates tokens with more than just two characters.

Compare the tokenization with WordPiece and word tokenization.

The word tokenization and WordPiece tokenization produced almost identical results. The only difference was that the WordPiece separated the apostrophe and the s in the word daughter's, resulting in three tokens (”daughter”, “‘”, “s”). Word tokenization solved it by using only two tokens (”daughter”, “‘s”). WordPiece tokenization also included special tokens [CLS] and [SEP] which are used by the BERT model.