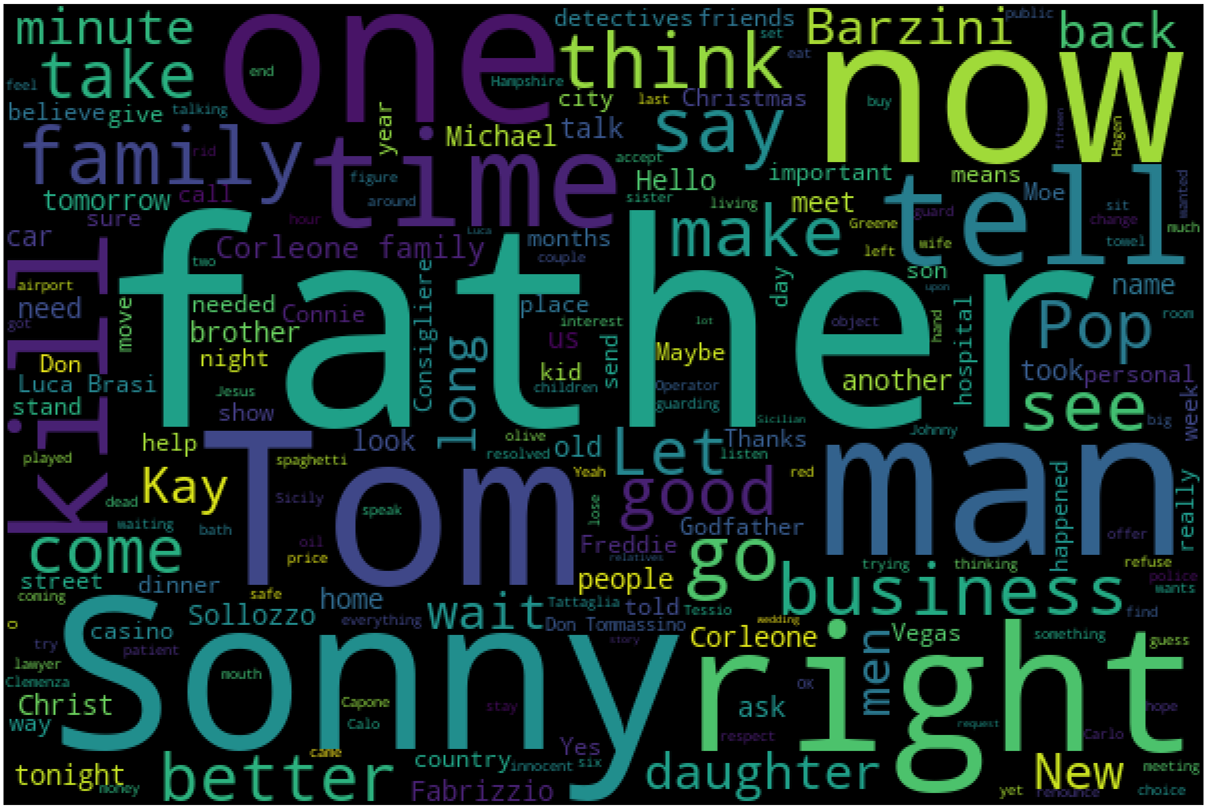
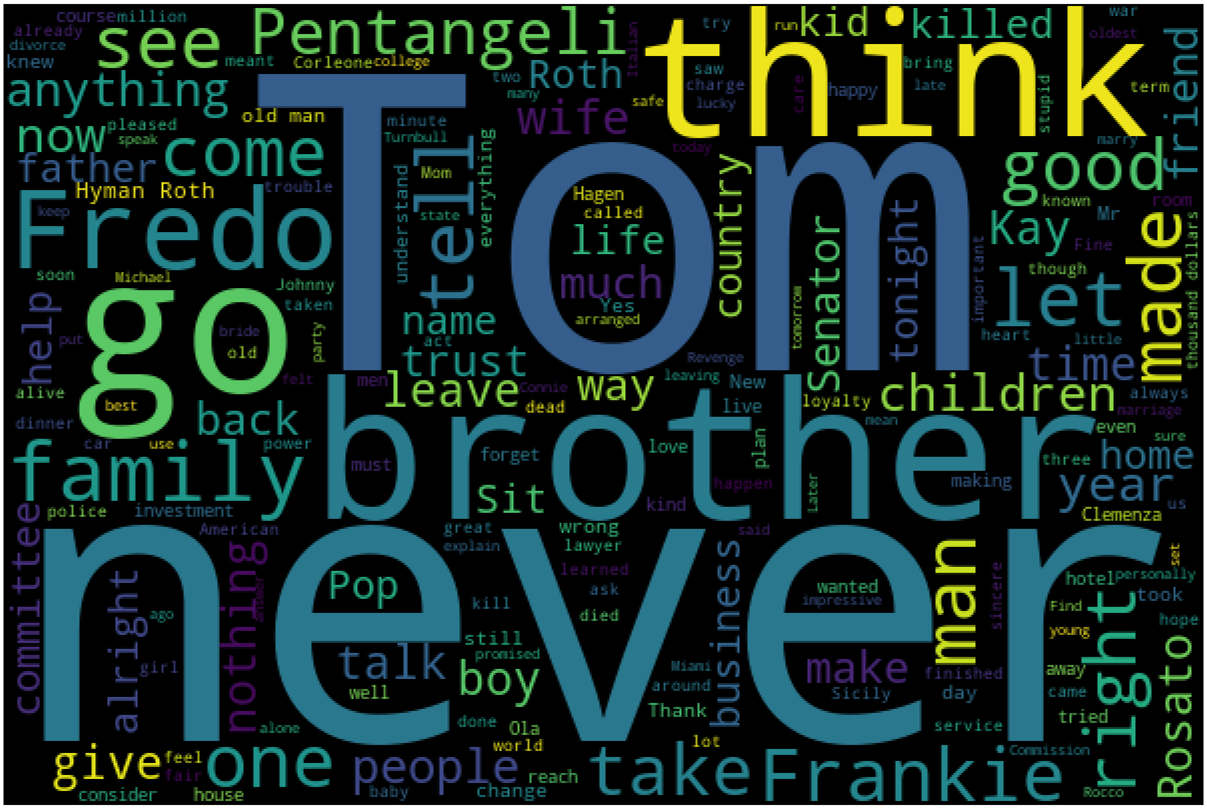
Godfather Pt. 1:



Godfather Pt. 2:



Nick Largey
Assignment 1
COS470: Text Mining

Task 2:

For these two graphs I used the WordCloud library in python which has a built in stopword list, which I extended to include "will", "know", "want", "going", "thing" and "things". I did this because the first plots produced had these words as the most frequent, and since they don't really tell us much about the dataset because of how they mainly rely on context within a sentence to provide truly beneficial information. I.E the sentence "I will find out where the thing is going, and let you know." And the sentence "You will find out where the thing is going and let me know." Have very different sentiments.

Looking at the plots, we can see that many of the most frequent words remain the same between the two movies, Tom is the most prominent character name that grows with the second movie, but we can also see the word "brother" grew significantly between the two films. Both growths, in this context, imply that these 2 subjects grew in importance within the overall story and plot since, typically, characters aren't mentioned more in films that aren't playing some sort of important role. This analysis should be taken with a grain of salt, however. If we consider "Corleone", the main characters last name, went down in frequency between the two we could believe that this character's role was reduced between the films, but knowing that he is the main character and that the word clouds are generated from his dialog, it could be that his importance grew since he is no longer introducing himself to the other characters.

It seems that these types of graphs very much rely on their context to provide beneficial information. I may have a hard time trying to deduce something like sentiment from a graph like this for something as abstract as dialog from a film, but if we were looking to find if, say, a product was liked by the public, using reviews as our data set, and seeing which words appeared the most frequently would be very beneficial.

Task 3:

Instead of breaking down a corpus of text into individual words and characters separated by whitespace and punctuation, the WordPiece tokenizer will begin by splitting the given text into separate words, creating a dictionary that contains the word and it's number of occurrences. It will then break down each of those words by letters, creating a new list that contains all of the characters used, and characters that have other characters preceding them, will account for this by adding '##' to the character in the list. It then calculates a frequency score for each possible matching of two of the characters in the list with the equation:

score=(freq_of_pair)/(freq_of_first_element×freq_of_second_element)

By using this equation, it will prioritize the largest possible combination of characters to add to the tokens list, instead of simply adding all the possible pairs and matches it finds. This learned vocabulary is then used to tokenize any input given to our model. For example, using the line below as our corpus, it would result in the following tokenizations.

Line: Because they know that no Sicilian will refuse a request on his daughter's wedding day

```
WordPiece: ['[PAD]', '[UNK]', '[CLS]', '[SEP]', '[MASK]', '##a', '##c',
'##d', '##e', '##f', '##g', '##h', '##i', '##l', '##n', '##o', '##q', '##r',
'##s', '##t', '##u', '##w', '##y', "'", 'B', 'S', 'a', 'd', 'h', 'k', 'n',
'o', 'r', 's', 't', 'w', '##ow', 'no', 'th', '##gh', '##ght', 'kn', 'know',
'on', '##ng', '##fu', '##qu', '##ught', '##us', '##fus', '##st', '##dd',
'##aus', '##caus', '##at', 'that', '##an', 'da', 'daught', 'day', 'Si',
'Sic', 'Sici', 'Sicil', 'Sicili', 'Sicilian', 'hi', 'his', '##ddi',
'##dding', 'wi', 'wil', 'will', 'Be', 'Becaus', 'Because', 'the', 'they',
're', 'refus', 'requ', 'refuse', 'reque', 'request', 'daughte', 'daughter',
'we', 'wedding']

WordTokenizer: ['Because', 'they', 'know', 'that', 'no', 'Sicilian', 'will',
'refuse', 'a', 'request', 'on', 'his', 'daughter', "'s", 'wedding', 'day']
```