Name: Nikita Raut **NetID: NBR1932**
Name: Anupam Tripathi **NetID: AAT1666**
Name: Jiuqi Xian **NetID: JXY9577**
Name: Akbar Imamov **NetID: AIE6401**
Name: Akash **NetID: ABS3211**

Group 4: Project 1
Topic: Comedy Movies and Drama Movies

Step 1

For this project, we selected wikipedia data since more number of topics were available to choose from and also text exactration process was easier. We had a few topics in mind, but after considering all factors, we selected movies of specific genres (comedy and drama) as our topic. We first constructed the appropriate queries for extracting the links from wikidata query tool. The very first issue that we faced was that the links were of wikidata.com instead of wikipedia. We then figured out a way to get redirected from that page to its corresponding wikipedia page. We used beautiful soup for extracting relevant data from the source code. We also have maintained a lot of metadata for all the movies in the form of dictionary, like 'directed by', 'produced by', 'release date' etc.

Step 2

In step 2, we first read the raw text file into a string. Then, we parsed the text to 1) strip html tags, 2) remove unwanted text eg: [10], 3) add <s> </s> tags and <quote> tags. We used regex to perform the above two operations and Beautiful Soup, Python package for parsing HTML. Then we created a function for adding spaces before and after parentheses, fullstops, comma, colon and 's, so that they can be tokenized. Then we replaced "-" with "@-@". The above two operations were performed using the replace function of python. Then we tokenized the text into words using NLTK package to create a python list. Next, we created a function to count the word frequencies and replaced the words which have frequencies less than 3 with <UNK>. Then, we split this list into train, test and valid data. However, when we directly took the first 250k words from the raw text file as test data, we only got words from the comedy genre. In order to get balanced data from both comedy and drama genre we created two separate files for each genre and chose 125k words each. Thus, we first created the test and valid sets. The words which remaining fell into the train set.

Step 3

In step 3 we constructed a list of tokens from the training text from step 2. Then we created a word-to-index mapping as a dictionary. Then we use this mapping to convert the text to integer arrays, and treat the unseen words in validation text and test text as "<UNK>" token and use the corresponding index of "<UNK>".

Step 4

For this phase we had to replace several keywords as well as numbers with special strings in training, testing and validation sets in order to make more robust dataset for future assignments. By using python regular expression library we declared 4-digit numbers as years in range of 1000-2999, and replaced with <<year>>. For any number(decimals) besides year, we replaced them with <<dec>>. We also replaced stopwords from the raw text files and in the end we end up tokenizing the outcome.