Project Progress Report
Names: Elizabeth Lin (el18@illinois.edu), Nirali Rai (narai2@illinois.edu)

GitHub: https://github.com/el18illinois/CourseProject-CS410-MovieBee

Code Functionality

An overview of the function of the code (i.e., what it does and what it can be used for).

The purpose of this program is to conduct sentiment analysis on movie scripts. By taking a movie script and deconstructing it into a bag of words, applying two methodologies, sentiment analysis can be performed to determine the tone, positive or negative, of the provided script. The two methods that are provided in this program include unigram sentiment analysis of the whole movie script and unigram sentiment analysis of subsections of the movie script.

The unigram sentiment analysis method is performed by taking a tokenized bag of words from a movie script, a single term in each token, and determining whether terms are positive or negative. The analysis would conclude if the movie script contained word choice that was more negative or positive. This differs from other forms of n-gram analysis as each token has a single term rather than phrases and combinations of words. This analysis is used to determine the tone of the scripts based on the words used. Using the program, we can see how negative or positive the word section is for a specific movie script. In addition, we apply sentiment analysis to subsections of the movie script. This allows for analysis of different parts of the movie, which, historically, can have different themes, allowing for comparison of the analysis outcome on a whole script and on the beginning, middle, or end of the movie script. This allows a more holistic view of how the sentiment changes throughout the movie phases. We can conduct this script for single or multiple movie script files, allowing comparison between movie scripts.

When we think about movies, for example, a Disney movie, the overall theme tends to be a happy or loving theme; but by conducting a sentiment analysis of the movie script, we are able to determine if the word choice matches that desired theme. Furthermore, we can see the progression of the sentiment throughout the movie. This analysis method can also be applied to other documents, such as songs, plays, speeches, etc.

Documentation

Documentation of how the software is implemented with sufficient detail so that others can have a basic understanding of your code for future extension or any further improvement.

The code can be broken down into three main sections: loading the word, clean up, and main.

The script first loads (load_words) the negative and positive words that are used to conduct the analysis. These words are pulled from two files, one containing negative terms and the other containing positive terms. These files can be updated given a selection of words that need to be added, updated, or removed. The two files are denoted as “negative-words.txt” and “positive-words.txt”; each file contains a single term on each line that falls into the file title’s category.

Next is cleaning documents; in this case, cleaning the movie scripts (clean_up). In this process, a movie script is taken and transformed into a data type that can be used for analysis. Its role is to clean up the movie script by first tokenizing and changing to lowercase, making everything uniform. Once the movie script is tokenized and lowercase, stop word filtering is done. During stop word filtering, all stop words are removed from the list of tokens.

The main (main) functionality occurs in the main, there two types of analysis are supported; unigram on the whole script and unigram on subsections of a movie script. During unigram sentiment analysis of the whole movie script, first, each movie script is taken and tokenized. Then the list of tokens is compared to the list of positive and negative words to calculate a score; the lower the negative number, the more negative terms are contained. Building on this, the movie script can be broken into subsections, and the unigram sentiment analysis is conducted on each subsection, resulting in values for the beginning, middle, or end.

The output is a table that is constructed with the following headers based on the analysis performed:

Script # (basic counter)	**Movie Script Name**	**Sentiment Score**

Running the Program

Documentation of the usage of the software including either documentation of usages of APIs or detailed instructions on how to install and run a software, whichever is applicable.

To run this program, first, you need to make sure you have cloned the code from Github and check the following files are in your directory. Note: all files will be included in the repo if you clone the repo correctly.

- main.py: which contains the source code
- data folder: directory named “data” that contains all the movie scripts
- positive-words.txt and negative-words.txt: list of words, each word on a single line for ranking movie script sentiment

For this project, you will need the metapy library. If you do not already have that run:

- *pip install metapy pytoml*

How to run the code:

1. ***git clone https://github.com/el18illinois/CourseProject-CS410-MovieBee.git***
2. On your command line/terminal, make sure to install anaconda from https://www.anaconda.com/
3. After you install anaconda, go to your terminal, input "**conda active base**"
4. Set up Python 3.5 with "**conda create -n py35_env python=3.5**”
5. Activate the environment with "**conda activate py35_env**"
6. Install necessary libraries with “**pip install metapy pytoml**”
7. ***python main.py***
    a. Pick what part of the script you want to run the analysis on (options are numbers from 1-4)

Output

The output is structured like a table, on the left you will have the title of the movie script and on the right, you will have the score.

- The score represents the difference in positive to negative terms in the movie script. The more negative the word means that negative terminology outweighed the positive vocabulary.

Example Output below:

```
(py35_env) elizabethlin@ElizabethsMBP4 CourseProject-CS410-MovieBee % python main.py
Welcome to the Movie Ranking Program!

ABOUT THE PROGRAM:
In this program we are parsing movie scripts to analyze the sentiment in terms of negative and positive words frequency and assigning a score
for each movie.

PROMPT:
Which part of each movies would you like to parse?
1) Beginning
2) Middle
3) Ending
4) Whole Movie

YOUR ANSWER: 2
LOADING RESULTS...

ALGORITHM RESULTS:
                          Movie  Score
0               FRANKENSTEIN.txt   -153
1                  STAR TREK.txt   -116
2               THORRAGNAROK.txt   -112
3              BATMAN BEGINS.txt   -106
4                  TOY STORY.txt    -96
5                   TWILIGHT.txt    -94
6                     AVATAR.txt    -89
7   HOW TO TRAIN YOUR DRAGON.txt    -88
8                       JAWS.txt    -73
9              JURASSIC PARK.txt    -70
10      THE BOURNE IDENTITY.txt    -70
11        MISSIONIMPOSSIBLE.txt    -63
12             KUNG FU PANDA.txt    -62
13     STAR WARS  EPISODE 1.txt    -54
14         INDEPENDENCE DAY.txt    -53
15            BLACK PANTHER.txt    -52
16               HAPPY FEET.txt    -52
17  PIRATES OF THE CARRIBEAN.txt    -51
18                     Mulan.txt    -49
19                        UP.txt    -48
20                   FROZEN.txt    -41
21       BACK TO THE FUTURE.txt    -39
22       THE LITTLE MERMAID.txt    -36
23             GHOSTBUSTERS.txt    -35
24     Beauty and the Beast.txt    -31
25              Ender's Game.txt    -30
26  THE CHRONICLES OF NARNIA.txt    -27
27                     SHREK.txt    -26
28                      Coco.txt    -19
29             MARY POPPINS.txt     27

INTERPRETING RESULTS:
You selected to analyze the  of our movie corpus. The output of our program indicates that out of our movie corpus of 30 movies, we have MARY
POPPINS.txt with the highest sentiment score of 27  which means the movie is overall the most positive. Inversely, the most movie script with
the most negative sentiment according our program is FRANKENSTEIN.txt with a sentiment score of -153 .
```

Team Contributions

Brief description of contribution of each team member in case of a multi-person team.

Nirali Rai

- Understanding what type of data needed
- Reading descriptions of the types of datasets (movie scripts) and how they are formatted
- Finding and collecting data files and example files
    - This included finding accurate scripts and converting them into a usable and realistic format for the program
    - Formatting and evaluating movie scripts
        - Ensuring variety of scripts testing
        - How to utilize scripts in sentiment analysis
- Documentation
    - How the code works and output

Elizabeth Lin

- Spec out the design of the algorithm and research upon applicable methods
- Implementing and testing the algorithm on collected datasets
    - Debugging metapy library and parsing of scripts
- Designing the program structure and communicating with Nirali about code changes and documentation
- Assist with verifying and editing documentation