“Earths Mightiest Heroes”

A Natural Language Processing Analysis

Mikayla J. Scott

Text Mining - IST 736
Syracuse University
March 23, 2023

Table of Contents

Introduction

Marvel's "*The Avengers"* is a movie franchise based on the Marvel comics, "*The Avengers"*. The comics, first debuted in 1963, written by Stan Lee are a collection of stories revolving around "Earths Mightiest Heroes" (Lee, 1963). The movie franchise is composed of four movies. The first in the series, "*The Avengers*", released on May 4, 2012, focuses on the origin stories of the team, and follows the team as they work to defeat Thor's brother Loki and regain control of the tesseract. The second movie, "*Age of Ultron*", released on May 1, 2015, the Avengers work to take down Ultron, a sentient artificial intelligence (AI) program written by Tony Stark and accidentally brought to life. The third and fourth movies, "*Infinity War*" and "*End Game*", the Avengers find themselves going head-to-head with Thanos, collecting infinity stones and battling complicated timelines. The Avengers films are among the highest rated movie franchises with an average Rotten Tomatoes score of 89% and grossing over 7.772 billion US dollars in box office sales.

This study focuses on the scripts for each of the Avenger movies. In all the movies combined there are 112 unique characters that have lines. There were 4,100 total lines that were used for exploratory data analysis and in the comparison of different classification model approaches. The models for this study attempt to classify both characters and movie titles by using lines from the movie scripts. This document will explain the results of exploratory data analysis performed on the movies and compare the performance of Support Vector Machine (SVM), Multinomial Naïve Bayes (MNB), Decision Tree Classifier and K-Means algorithms on the movie scripts.

Data

Collection

The data for this study was composed of lines that were mined from the scripts of each of the movies. The movie scripts originated from the free movie script website, movies.fandom.com. The scripts were in a text format that followed this string pattern:

"TONY: What? Rock of Ages giving up so easily?"

Where the name of the character is proceeded by a semi-colon, and the line said by that character. Text from each of the movies was copied from the websites into a text (txt) file and saved. Those text files were then converted into comma separated value (csv) files using the semi-colon as the delimitator. This process put the entire script into a format that could be easily interpreted as a *Pandas* data frame later in the experiment. After bringing the text into the csv file the format resembled the table below:

Character	***Line***
TONY	What? Rock of Ages giving up so easily?
STEVE	I don't remember it being ever that easy. This guy packs a wallop.
TONY	Still, you are pretty spry, for an older fellow. What's your thing? Pilates?

Table 1: Script Data Frame

The csv file of the characters and their lines for each of the movies were combined in a *Pandas* data frame for further text processing in a *Python* (3.9) environment. Read more about the data processing and cleaning phase in the cleaning section.

Cleaning

The csv files for each of the movies were read into 4 different data frames, one for each movie. After the data was read in, an additional column was added, "*movieNum*". *movieNum* is a numeric column scaling from 1 to 4, where the numeric variable indicates the number movie (1: *The Avengers*, 2: *Age of Ultron*, 3: *Infinity War*, 4: *End Gam*e) in the series that the line corresponds to. Data frames were then concatenated, so all four movies were in the same data frame.

There were several steps that the "Character" and "Line" columns had to go through to be considered clean and ready for analysis. Scene notes that were included in some of the lines in the *Line* column were removed first. Scene notes are actions, emotions or feelings that the actor/actress is supposed to portray as they deliver the line. The scene notes in the script were written in between two brackets ([ ]). See an example of a line with scene notes below:

"[to Cap] You want me to put the hammer down?"

Using the regular expression pattern "\(.*?\)|\[.*?\]", everything in-between the brackets and the brackets themselves were removed from the line and replaced with no characters. After the completion of the first phase of processing the same line that was used as an example above reads as:

"You want me to put the hammer down?"

After the removal of scene notes, lines then went through standard text pre-processing phases. All lines were made lower case, and special characters were removed. The lines were then tokenized, and stop words were removed using the Natural Language Toolkit's (NLTK) stop word list. Additional words were added to the stop word list, see *Appendix Table 1: Additional Stop Words* for the appended stop words. The column *wordCount*, the count of words in the tokenized line list, was also generated during this phase. After stop words were removed the tokenized list was put back into a string and stored in the column *cleanedLines*. The line from the previous two examples post processing would look like this:

"hammer down"

The "Character" column had some additional issues. In the scripts there are multiple characters whose names are written in many forms. For example, Bruce Banner, Banner, Bruce, Doctor Banner, The Hulk, and HULK are all the same character, just represented in various ways. This is true for several of main characters. To maintain consistency across the dataset, superheroes were represented as their street names, all the variations of Iron Man were changed to Tony Stark, variations of Captain America were changed to Steve Rogers, and so on. See an example of the data frame after processing below:

Character	*Line*	*movieNum*	*wordCount*	*cleanedLines*
Thor	You want me to put the hammer down?	1	8	hammer down

Table 2: Script Data Frame Post-Processing

Exploratory Data Analysis

Of the 4,126 Lines in the dataset there are 112 different characters represented, and 39,893 words spoken throughout the four films. This section is dedicated to exploratory data analysis of the corpus and the corresponding variables.

Line Count

Line count for each of the characters was determined by grouping the data set by *Character* and then counting each of the rows in the groups. It was further aggregated by movie (*movieNum)* to show the number of lines for each of the characters for each of the movies. The character with the most cumulative lines throughout the entire franchise is Tony Stark, followed by Steve Rogers, Bruce Banner, Thor, Natasha, Nick Fury and Clint Barton. See the chart below for a depiction of the distribution of lines throughout the movies:

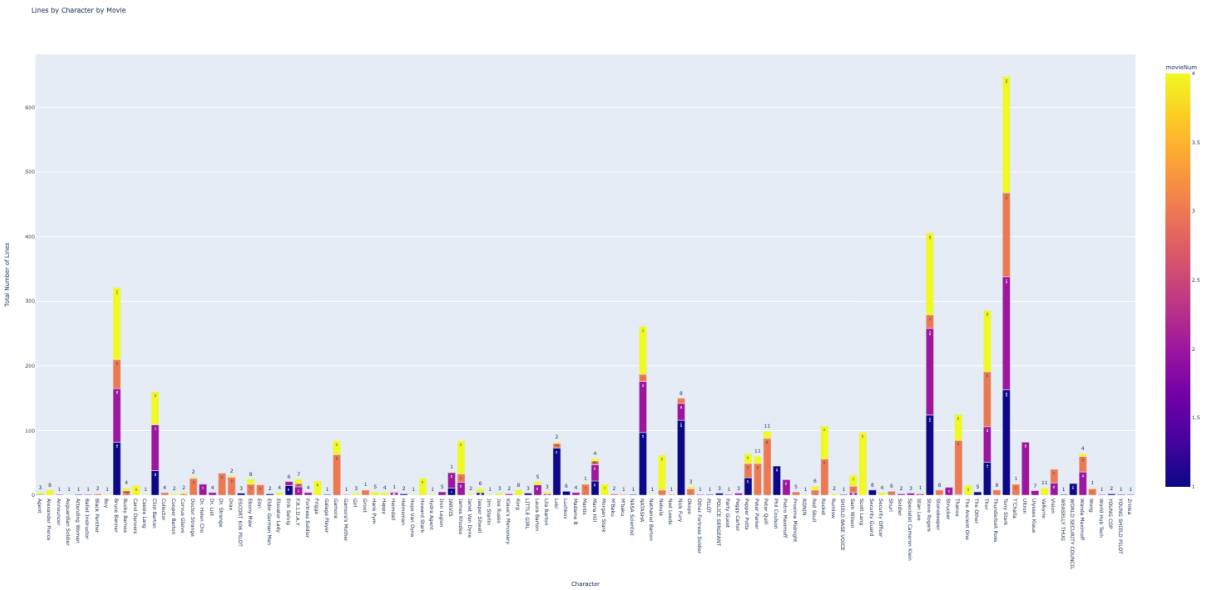

Figure 1: Lines by Character by Movie

Word Statistics

The first analysis on words was performed to determine the number of words in each of the movies, and further aggregated to show the number of words said by each character. The results were calculated by grouping the data frame by *movieNum* and *Character*, then summing the *wordCount* column for each of the rows. There were 39,893 words said in all the movies combined, 9,219 from "*The Avengers*", 10,209, from "*Age of Ultron*", 9,187, from "*Infinity War*" and 11,278 from "*End Game*". In all four movies Tony Stark says the most words,

however; in each of the sequels, the character that says the second most words change. In the first movie it is Nick Fury, second is Ultron, the third is Thanos and fourth is Thor. See the diagram below for a breakdown of the word count by character by movie:

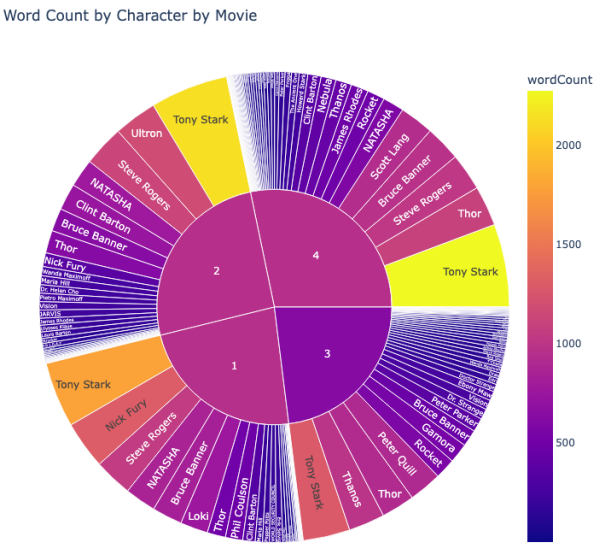

Figure 2: Lines by Character by Movie

The most popular word by frequency for all the movies was also calculated. Unlike the data in the previous two analysis cases, the frequency of words was determined using the cleaned lines. Stop words were removed to reduce noise, and to show more prominent terms across the dataset. The top 50 words for all the movies are shown in the graph below:

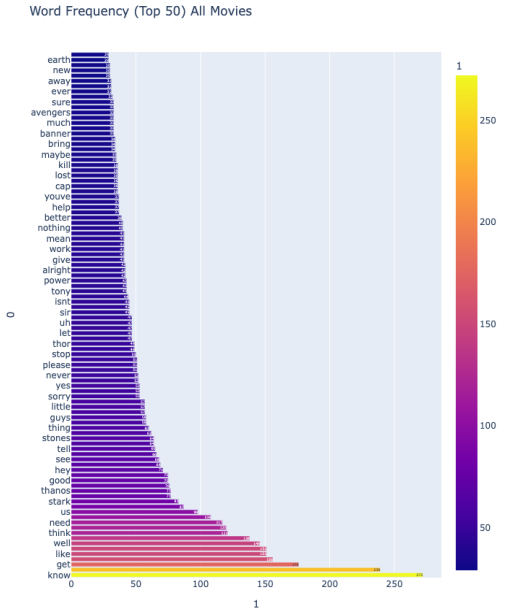

Figure 3: Word Frequency (Top 50) All Movies

Sentiment Analysis

Sentiment analysis using NLTK's Vader Lexicon was utilized to determine the polarity of each of the characters throughout the movie franchise. Numeric columns, *Pos*, *Neu*, *Neg* and *Compound* were added to the data frame for each line. Average scores were then calculated for each of the characters for all the movies by grouping the data frame by *Character* and *movieNum* and taking the mean of the rows of each of the groups. The compound polarity scores work on a scale from 1 to -1 where 1 is very positive and -1 is very negative. There are characters that stay rather stagnant throughout all the movies, like Natasha, and there are characters that change dramatically throughout the movies, like Thor. For legibility reasons, only the six original Avengers are shown in the graphic below. The graphic depicts the average compound sentiment score for each of the characters over the four movies.

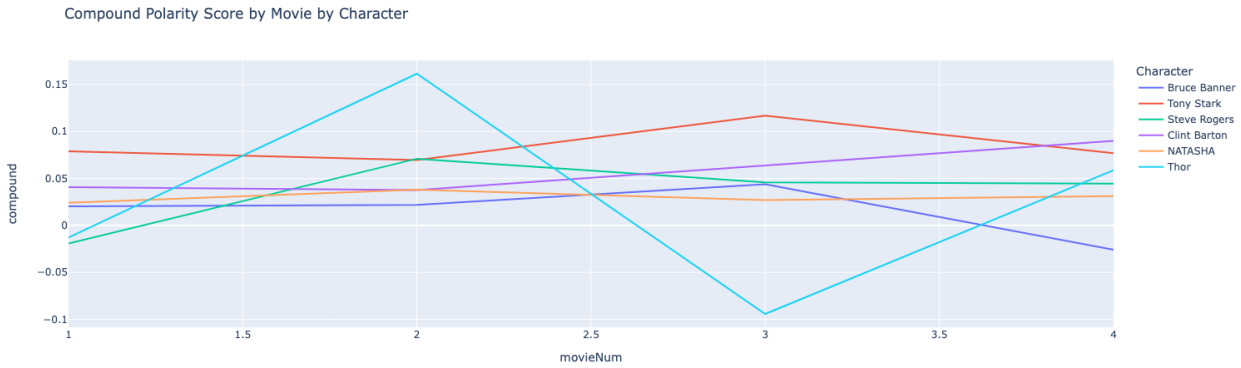

Figure 4: Compound Polarity Score by Movie by Character

Network Analysis

A network of characters was created to see the connections between characters across movies. Each of the movies is represented by a different color path, the red paths correspond to "*The Avengers*", blue paths are "*Age of Ultron*", green paths are "*Infinity War*", and the grey paths are "*End Game*". Character nodes that fall to the center of the network are characters that appear in all four movies. The original team, Iron Man, Thor, Captain America, Hawkeye, Black Widow and the Hulk all fall into this category. The centrality of these specific characters is also supported by their degree centrality score. Each of the original avengers scored above .033, where most characters degree centrality measured around 0.015. Other notable characters include Maria Hill, JARVIS and Nick Fury. Closeness centrality, betweenness centrality and page rank were also calculated.

For all the movies there is an average of 18 unique character appearances, where a unique character appearance is defined as a character who only appears in a single movie. Notable characters that fall into this movie are Ultron, and Phil Coulson. The largest overlap of characters between two movies come from "*Infinity War*" and "*End Game*". In addition to the original Avengers team there are 15 characters that appear in both the third and fourth movies. Some notable characters in this overlap are Peter Parker, Doctor Strange, Groot, Gamora, and Peter Quill. See the network of characters in the network graph below:

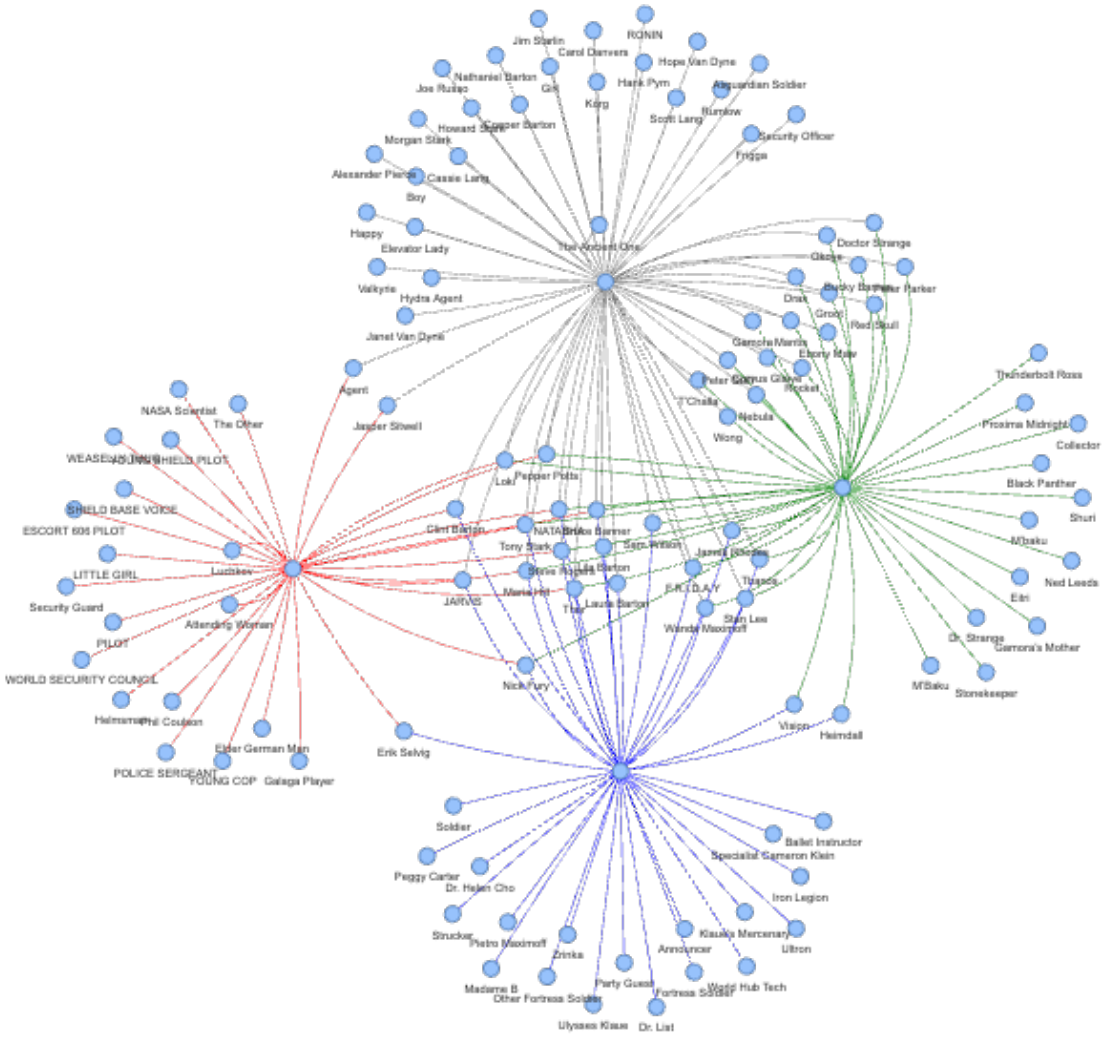

Figure 5: Character Network Graph

Modeling Methods

The goal of modeling this dataset was to determine the best classification model to classify lines by the character and by the movie that they originated from. This section will describe the models that were implemented, the features that were used and different text vectorizers that were utilized.

There were four different kind of classification models that were compared for this study, multinomial naïve bayes (MNB), support vector machine (SVM), decision tree classifier (DT) and K-Means. Each model was trained to either predict a movie, or to predict a character based on the lines that were fed into it. For the MNB, SVM and DT models, there were four different versions of each model trained and tested. The different versions of the models were trained using a different SKLearn text vectorizers to prepare the lines for processing. The four

vectorizers that were used is as follows: unigram count vectorizer, unigram Boolean vectorizer, Gram-12 count vectorizer and term frequency inverse document frequency (TFIDF) count vectorizer. Unigram count vectorizers tokenize the text in by single word and creates a matrix of token counts, the Boolean vectorizer does the same thing with the exception that all non-zero counts are set to 1. The Gram 12 vectorizer works with n-grams, and the TFIDF vectorizer produces a matrix with a measurement of originality within the document. Where count vectorizers focus on the frequency of terms in a document, TFIDF is calculated by calculating the term frequency of a word and multiplying it by the inverse document frequency.

In total there were 29 models trained and tested. See the *Results* section for the experiments ran on the models and the results produced.

Results

Character Classification - All Characters

The first models were trained with all the characters. With so many characters the models struggled to classify any of the characters correctly, this was true across all vectorizers and different algorithms that were used. There was a lot of confusion with characters that only had a few lines in the movies. Those characters were continuously classified as characters that have the most lines throughout the movie. Characters like Tony Stark, Steve Rogers and Thor were the characters that were predicted the most. The model that performed the best out of this group was the TFIDF MNB model. The results of the models can be seen in the table below:

Model	***Type***	***Accuracy***	***Model Goal***	***Features***	***Vectorizer***
1	MNB	0.1663	Predict Character	ALL characters	Unigram Count Vectorizer
2	MNB	0.168	Predict Character	ALL characters	Boolean Count Vectorizer
3	MNB	0.154281	Predict Character	ALL characters	Gram 12 Count Vectorizer
4	MNB	0.169628	Predict Character	ALL characters	TFIDF Count Vectorizer
5	SVM	0.149434	Predict Character	ALL characters	Unigram Count Vectorizer
6	SVM	0.15105	Predict Character	ALL characters	Boolean Count Vectorizer
7	SVM	0.147011	Predict Character	ALL characters	Gram 12 Count Vectorizer
8	SVM	0.164782	Predict Character	ALL characters	TFIDF Count Vectorizer

Table 3: Character Classification – All Characters – Model Results

Character Classification – Original Avengers

In result of the accuracy scores being so low in the last round of model testing, this round of modeling only uses the original characters for prediction. This includes Thor, Tony Stark, Natasha, Clint Barton, Bruce Banner and Steve Rogers. A subset of the data frame containing

only these characters was used as the features for this round of models. In addition to the MNB and SVM models used in round one, in this round of testing decision tree classifiers and a K-Means model were trained and tested in effort to produce better accuracy scores. The models continued struggling to classify characters by their line vectors. Most of the lines for each of the characters, if not classified correctly, were classified as Tony Stark. Bigrams continued to produce the worst results across all modeling methods. See the results for the MNB, SVM and DT models *Table 4*:

Model	***Type***	***Accuracy***	***Model Goal***	***Features***	***Vectorizer***
9	MNB	0.30149	Predict Character	Original Team	Unigram Count Vectorizer
10	MNB	0.30298	Predict Character	Original Team	Boolean Count Vectorizer
11	MNB	0.307	Predict Character	Original Team	Gram 12 Count Vectorizer
12	MNB	0.31	Predict Character	Original Team	TFIDF Count Vectorizer
13	SVM	0.2552238	Predict Character	Original Team	Unigram Count Vectorizer
14	SVM	0.2626	Predict Character	Original Team	Boolean Count Vectorizer
15	SVM	0.307	Predict Character	Original Team	Gram 12 Count Vectorizer
16	SVM	0.2582089	Predict Character	Original Team	TFIDF Count Vectorizer
17	DT	0.31642	Predict Character	Original Team	Unigram Count Vectorizer
18	DT	0.31194	Predict Character	Original Team	Unigram Boolean Vectorizer
19	DT	0.3044777	Predict Character	Original Team	Gram 12 Count Vectorizer
20	DT	0.308955	Predict Character	Original Team	TFIDF Count Vectorizer

Table 4: Character Classification – Original Team – Model Results

The K-Means model performed worse than any of the models. In every iteration of training, it continued to cluster almost all the 1,562 lines in the training vector together. See the clustering iterations in the table below:

	Cluster 0	***Cluster 1***	***Cluster 2***	***Cluster 3***	***Cluster 4***	***Cluster 5***	***Cluster 6***
1	6	1529	12	1	8	1	5
2	24	28	1486	1	8	14	1
3	1407	96	1	23	1	20	14
4	12	1342	49	24	91	21	23
5	1534	1	1	1	1	23	1

Table 5: K-Means Iterations

Movie Classification

For movie classification the goal was to identify the movie by the line that was said. The models in this round of testing performed a little bit better than those in the previous two rounds of testing; however, results still left more to be desired. The model that produced the best results was the MNB model that was trained and tested with unigram Boolean vectors. There were several words that were distinct indicators of a specific movie, a few of those words are listed below:

The Avengers	***Age of Ultron***	***Infinity War***	***End Game***
Loki	avengers	Soul	Father
Tesseract	cap	Gamora	Quantum
Stark	Ultron	Dead	Sorry
Suit	Strucker	Universe	Stone
Barton	Jarvis	Thanos	Time

Table 6: Indication Words

The results of the different models and their accuracies can be seen in the table below:

Model	***Type***	***Accuracy***	***Model Goal***	***Features***	***Vectorizer***
22	SVM	0.384	Predict Movie	All Movies	Unigram Count Vectorizer
23	SVM	0.394	Predict Movie	All Movies	Unigram Boolean Vectorizer
24	SVM	0.3045	Predict Movie	All Movies	Gram 12 Count Vectorizer
25	SVM	0.4	Predict Movie	All Movies	TFIDF Count Vectorizer
26	MNB	0.41	Predict Movie	All Movies	Unigram Count Vectorizer
27	MNB	0.42	Predict Movie	All Movies	Unigram Boolean Vectorizer
28	MNB	0.31	Predict Movie	All Movies	Gram 12 Count Vectorizer
29	MNB	0.41	Predict Movie	All Movies	TFIDF Count Vectorizer

Table 7: K-Means Iterations

All the models and their accuracies can be seen in the graph below:

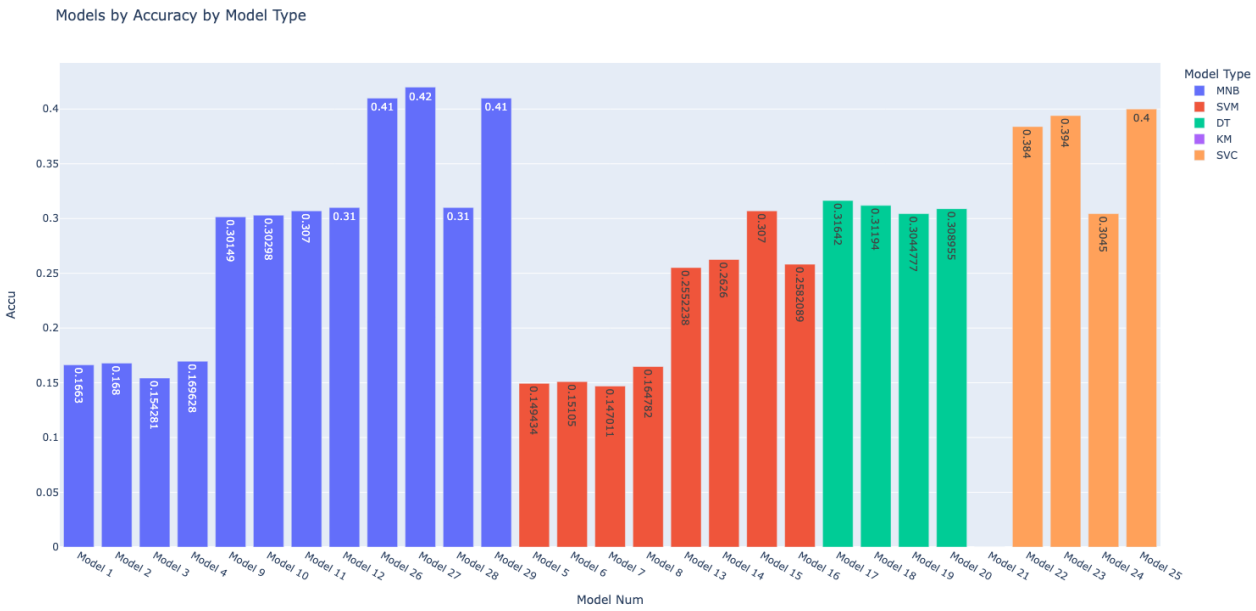

Figure 6: Accuracy of All Models

Conclusion and Recommendations

The accuracy results across all the models were considerably low. After reviewing the results of the models there are a few attributes of the dataset that could potentially contribute to the low scores. The dataset is imbalanced. There are characters that have roughly triple the number of lines that other characters do, even when the dataset is subset into just the main characters. In result, the models misclassify a lot of the lines for lines that belong to the majority class, in many cases this is Tony Stark. For further development, character lines should be under sampled or over sampled.

The movies take place over a seven-year span and fall within the plot line of the first three phases of the Marvel universe. The main characters have their own spin off films and appear in other movies that contribute to the Marvel universe timeline. Those movies ultimately affect the plot line of the *Avenger's* movies. To get a somewhat accurate representation of the sentiment of the main characters throughout time, other movies should be taken into consideration.

With the addition of more films, classification results could also increase. In the films the characters continuously talk about the same subject matter. With the removal of stop words and the cleaning process, a lot of the lines from each of the characters start to be copies of each other. If all the characters are saying the same thing, there is no differentiation for the models to pick up on. By adding more lines from additional Marvel films, the models might have a better chance at spotting differentiation in the way each of the characters speak.

Additional variables could be added to the dataset for a more in-depth look into sentiment analysis. Instead of looking at the movie scripts it might be interesting to look at the transcriptions of the movies and gather the timestamp of when each line was said. Once timestamps are collected, calculate the sentiment of each of the lines, and plot the results across

the duration of the movie. Then compare the visuals with other films in the Marvel franchise. It might be interesting to see if there are patterns that the typical Marvel movie follows, then aggregate by character, and see what kind of emotional journey the characters have been on throughout the films.

Appendix

I've	Really	Come	That's	Didn't
Gonna	He's	His	You're	You
I'm	They're	He's	Oh	Didn't
That's	Don't	Way	Got	Yeah
Okay	I'll	There's	Gotta	What's

Appendix Table 1: Additional Stop Words