Star Wars Analysis Technical Report

Miles Mercer

# Introduction:

Star Wars has, for a long time, been a significant interest of mine. I’m a major fan of many different elements of the franchise. One thing that has been particularly interesting to me is how stories have grown and branched out from the original trilogy. I recently had a conversation with a literature professor named Dr. Tredennick, who argued that the original Star Wars was, at its core, an incredibly safe and even algorithmic movie. She claimed that the success of this movie prompted a wave of sci-fi movies afterward that were equally algorithmic and somewhat limiting for the film industry as a whole. I found this claim interesting, and wanted to see if I could use natural language processing to represent this “algorithmic” plot curve visually.

The dataset for this project is called “Star Wars Movie Scripts”. It can be found on Kaggle at this [link](https://www.kaggle.com/datasets/xvivancos/star-wars-movie-scripts). The dataset contains three CSV files, one for each of the original three Star Wars movies. These files come directly from the original movie scripts. There are three columns in each file: “line”, “character”, and “dialogue”. The file for Episode IV has 1010 lines, not including the header. The one for Episode V has 839. The one for Episode VI has 674. In addition to these three files, I created my own CSV file classifying each character as a protagonist, antagonist, or neither. There weren’t too many characters in the scripts, so this was a fairly easy task to complete manually. Once I completed my basic sentiment analysis, using these sentiments to classify characters as protagonists and antagonists became the goal of my classification work.

At its most direct implications, this project doesn’t appear too impactful. The end goal has always been sentiment analysis on Star Wars scripts. That isn’t to say, however, that the work might not have other applications. In my analysis, I did a significant amount of work developing algorithms to analyze the relationships between the sentiments a character expresses and those that they receive. While my use of this information was fairly simple, I could see it having more complex applications. Say, for instance, a social media company wanted to develop an algorithm to detect and flag cyberbullying among underage users of its platform. Having knowledge of the sentiments people are expressing on average contrasted with those they are met with in response could be valuable information in detecting bullying relationships. As for now, though, my goal is purely to analyze Star Wars movies.

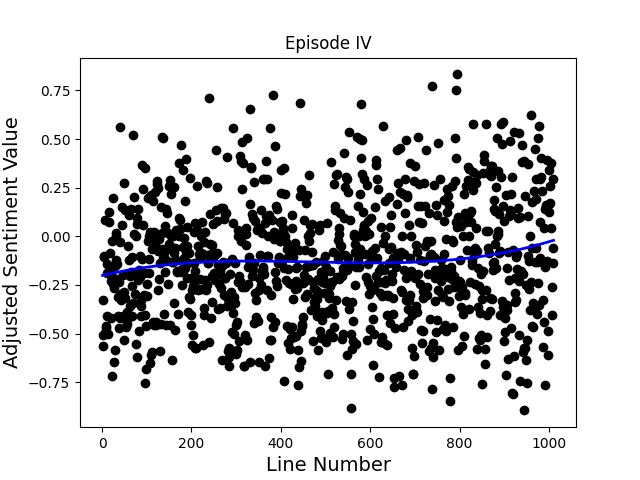
# Data Analysis:

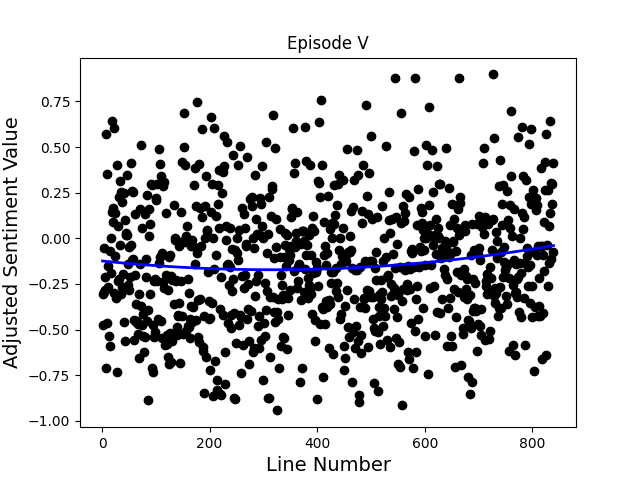
I was fortunate in my early analysis to have data that was already clean. In over 2500 lines of dialogue between the three movies, there was not a single missing entry. I did find a few typos here and there, but they didn’t appear to have any discernible effect on the performance of the sentiment analysis model. For my data analysis, I wanted to begin by evaluating the expressed sentiment for each line in each of the three scripts. To accomplish this, I enlisted the help of my friend and colleague Gabe DiMartino. He was, at the time, working on developing his own AI assistant. For this project, he had written some Python code to identify the intents of prompts to the AI assistant. I was able to gain access to this code, which can all be found in this [GitHub repository](https://github.com/Macbee280/CrimsonCode2023). This code creates a Rasa NLU (Natural Language Understanding) model based on data in a JSON file and uses that model to train a Spacy intent classification model which can, in turn, identify the intent of a given string. Because Rasa NLU is not quite as popular as the other libraries I used, it isn’t updated as often. To deal with this, I conducted all of my sentiment analysis with a Python 3.7 virtual environment instead of 3.10.

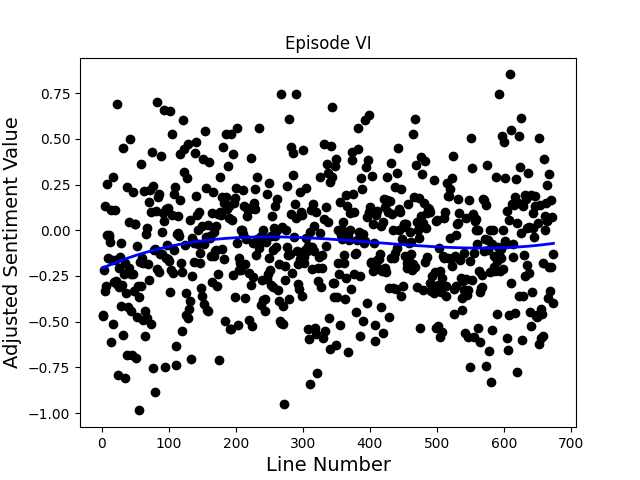
Gabe advised me that I would get the best performance from the resulting model if I made my own training JSON file with lines directly from the movies. To make this file, I wrote a program in C++ that would take user input for a line, the sentiment of that line, and any entities referenced in the line, then update the JSON file with the information. I gave this program lines from the first movie and half of the second, equally split between positive and negative lines. In total, there were 40 lines classified as “Positive” and 40 classified as “Negative”. I then used this file to make a Rasa NLU model and then that file was used along with a common English library to train the Spacy model. The model could now identify the “intent” of a line, which was really just its sentiment, and the entities in the line as well. I then used the confidences of the model’s intent assessments to create numerical sentiment values. If the model identified the intent of a line as positive, the resulting value would be the confidence minus 0.5, multiplied by 2. If it identified it as negative, I multiplied it by -2 instead.

The sentiment values and lists of entities were then added to the script CSV files. Now, in addition to line, character, and dialogue, the files had columns labeled “sentiment” and “entities”. The entity detection, though, didn’t prove consistent enough to use in analysis, so it was never referenced beyond this point. I was still able to complete much of my desired analysis without this information.

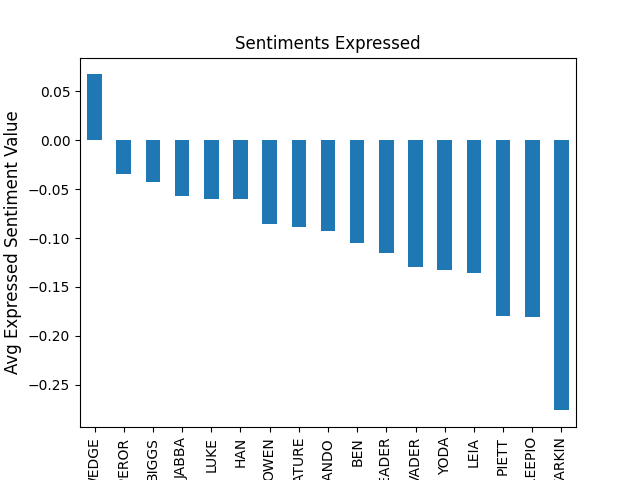
I began my data analysis with some simple regressions of the sentiment value with respect to line number. For each movie, I created a scatter plot with this data, made some quick observations of the behavior of the graph, and applied an appropriate polynomial regression. Episodes IV and VI each received cubic (degree 3) regressions, but a quadratic (degree 4) regression seemed more appropriate for Episode V. These regression lines were then drawn over the data. These are the resulting graphs and equations:



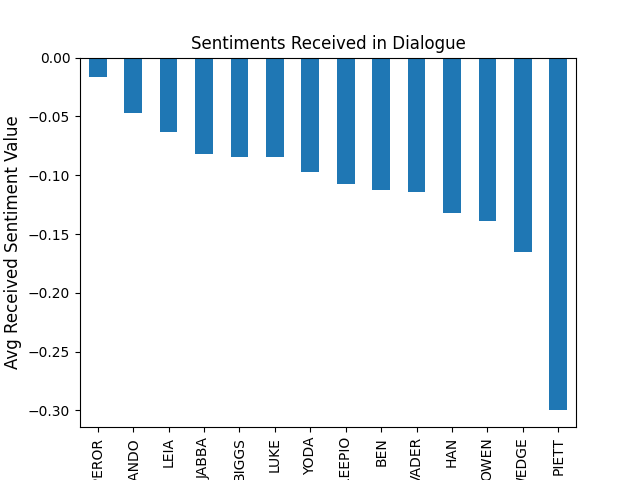




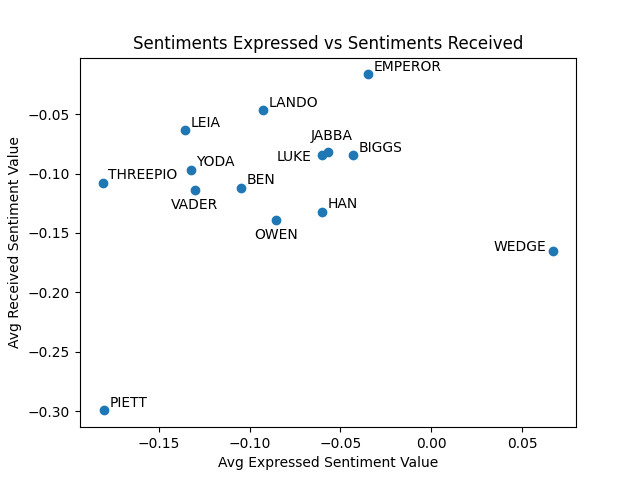
I then calculated the average sentiment for each character. Of course, there are a lot of characters in the original three Star Wars movies, and very few of them have a significant number of lines. So, I decided to only look at characters with 20 or more lines between the three movies. I then sorted them in descending order by average sentiment and created a bar graph. This is that graph:



I also wrote an algorithm that cuts the movies up into separate dialogues. Whenever two characters spoke back-and-forth for at least 4 lines, that was identified as a dialogue. Using these separate dialogues, I could now analyze not only the sentiments characters were expressing, but also the ones they were receiving. Based on the sentiments expressed by the other character in any dialogue a given character appeared in, I could calculate the average sentiment the character received. Here is the bar graph for the average sentiments received by different characters:



From here, I could simply turn the two series used to store the average expressed and received sentiments into data frames and then merge them together. I performed an inner join based on the “Character” column, resulting in a shorter data frame, but one that only contained characters who appeared in both of the above graphs. Once I had this data frame. I created one more scatter plot with each character’s average received sentiment with respect to the average expressed sentiment. This is that graph:



Before this analysis, I might have predicted that characters who expressed more positive sentiments would receive positive ones in return. In part, this turned out to be true. The graph does appear to have a vaguely positive trend. I think what it shows more clearly, though, is that all characters tend toward negativity. I suppose that isn’t particularly surprising. Negative sentiments are needed to present conflict and drive the plot. In fact, in a series called Star *Wars*, perhaps the trend toward negativity in dialogue shouldn’t be surprising at all. I think what makes this graph more interesting is how the data reflects the characters as we remember them. For instance, Piett was written as a constant nuisance to Darth Vader. He’s designed to be annoying. This is very clearly shown in the data. He receives by far the most negative sentiments and expresses very negative ones as well. Parallels between observable character traits and sentiment data are everywhere in this graph.

# Classification Results:

As mentioned earlier, the goal of my classification work was to classify whether the speaker of a line is a protagonist or an antagonist based on the sentiment of the line. Initially, there were 1907 lines between the three movies where the speaker was a protagonist and only 324 where the speaker was an antagonist. For reference, that’s about an 85% protagonist, 15% antagonist split. To solve this issue, I removed all of the instances where the speaker was classified as “Neither” and then sampled from the protagonist lines without replacement until there was an equal number of lines from protagonists and antagonists.

I hypothesized that lines with higher sentiment values would have a higher likelihood of having been spoken by a protagonist. I used my randomly balanced script and performed a train-test split with 20% of the lines reserved for testing. I then initialized a K Nearest Neighbors (KNN) Classifier model and fit it to the training data. When I tested its accuracy on the test data, however, its accuracy was only 0.55. This was barely better than if I had made a model that randomly guessed “Protagonist” or “Antagonist”.

I also tried using a Decision Tree classifier to see if it would perform any better, but it didn’t. In fact, it scored a 0.54 on the test data as opposed to the KNN classifier’s 0.55. Clearly, this data was not showing the correlation I thought it would.

So, what went wrong? One would think that protagonists would naturally express more positive emotions and antagonists more negative ones. When we try to test this hypothesis, however, it doesn’t hold up. I have to believe this has everything to do with the difference between static and dynamic characters. When I looked at the kinds of sentiments protagonists were expressing, they were all over the place, whereas antagonists didn’t display the same spread. This, to me, indicated that the protagonists were just written as more dynamic characters. They’re on screen more, so they have more time to display a wider range of emotions and grow as characters. The same exact character might be expressing entirely different sentiments at the beginning and end of a movie.

Across all the protagonists in a movie, this means that protagonists are covering a broad range of emotions, which doesn’t leave much of an “antagonist range” in the sentiment values. There’s no way to fit a classification model to sentiment alone and get accurate predictions of whether a speaker is a protagonist or antagonist because protagonists are expressing too many different sentiments to discern.

# Conclusion:

In conclusion, while I wasn’t able to produce accurate protagonist and antagonist classifications based on sentiment analysis of lines, this project produced some interesting results. I visualized the shift in sentiment over the course of a movie and the relationship between the average sentiment a character expresses and the average sentiment they receive in dialogue. I was able to identify parallels between qualitative observations of character traits and quantitative trends in their sentiment data.

In regard to the initial critique of Star Wars from a literature professor that, in part, inspired this project, my analysis produced some interesting evidence to use in response. Perhaps the case could be made that Star Wars is algorithmic because it follows a discernible plot line. I suppose that’s what I was analyzing when I put regressions on the sentiment of the movies over time. If it were true that the movies lack dynamic characters, however, my classification models probably would have performed a little better. I would need more evidence to back up this claim, but I believe it is because the protagonists displayed such a wide range of emotion and sentiment that I couldn’t find any ranges of sentiment that were predictive of characters being protagonists at all.

I previously described how my work could be used in social media platforms. Having now described at length the project’s progress, there are some important takeaways. It has become clear to me that sentiment alone is not adequate to make predictions about characteristics. If one wanted to make accurate assessments of character traits or relationships from sample interactions, they would need to look at a broader range of intent than sentiment alone. Sentiment doesn’t tell the whole story.