Rajat Sethi

ECE 4990

Project Write-Up – IMDb Natural Language Processing

**Introduction:**

The internet stores a large database of movie reviews and ratings on imdb.com. On this website, the general public can publish their own opinions about their recently watched shows and movies. The main goal of IMDb is to inform viewers about media they should watch or avoid based on the critiques of others. In addition, these reviews allow content creators to reflect on their mistakes and produce better material in the future. By mining the reviews and analyzing them, my project seeks to create new stories based on what aspects the public has highlighted in their opinions.

**Methodology – Data Collection:**

I tried a few methods when collecting data. At first, I looked at IMDb’s publicly available datasets. While they’re free to download and had reviews, they did not have the specific ratings. As such, I would not have had any labels to train the data with.

My next approach was through web scraping. By sending requests to the IMDb website, I could parse the page source for each review. At first, I tried using the AIOHTTP Python package to asynchronously request a couple thousand reviews. However, IMDb notices when too many requests are being sent at the same time, and the responses are replaced with nonsense.

Finally, I attempted selenium, a software that automatically opens browsers and runs requests. Using a headless Firefox/Geckodriver, I was able to send 10,000 requests to IMDb and pull their source code. Around 6,000 of them return valid responses. Once I received the source code, I used a regular expression to pull out both the rating and the review. This data would subsequently be stored in a text file for further processing.

**Methodology – Data Processing:**

When text is being analyzed, it had to be cleansed in many ways. Punctuation, capitalizations, conjunctions, and misspellings must be removed. In addition, common filler words (i.e., the, an, it, her, etc.) also need to be removed because they do not capture the “essence” of the review.

Once the review is stripped of its non-essential aspects, the words are then quantified so that calculations can be made. First, the words are vectorized, meaning that they are translated into a large matrix of numbers. These numbers represent a word’s characteristics, and they can also be used in arithmetic. For example:

“king” = “queen” – “woman” + “man”

After that, the words are also quantified by their frequency in the sentence, using what’s known as a TF-IDF Transformer. This allows repeated words to have more of a presence in the final analysis.

**Results – Data Modeling:**

Modeling the data can be accomplished in several ways. Due to time constraints, I had to use a simplistic logistic regression model, which yielded a 40% accuracy rating for the testing set. This portion of the research will be expanded upon in future efforts, particularly with the use of TensorFlow/Keras. I had attempted a sequential model with dense layers and a softmax activation function, but I had received a few errors and I did not have enough time to debug it.

**Future Work:**

The main point of improvement is the model. By developing a neural network with the softmax function, reviews can be categorized from 1-10 stars. This includes reviews that “do not exist,” or rather, the synopses for unreleased stories. This will eventually lead into Natural Language Generation, which should create ideal and non-ideal reviews, thus generating ideas for new media.

For the security aspect of this class, I would like to look into data poisoning. For example, giving a very negative review but then also rating it a 10/10. Reviews like that will most certainly cloud the accuracy of the model, and I would like to find solutions to such poisoning.