Sentiment Analysis of Song Lyrics

There are millions of songs in the world, each one with unique melodies and lyrics. These lyrics give way to the message the song wishes to convey and the feelings the song wants to evoke. I enjoy listening to music because of this diversity across songs in their sound and emotions they arouse. Pop music is usually light and upbeat with themes of romance, whereas rock typically consists of heavier instrumentals and vocals and darker themes. I listen to a wide variety of music, and it would be interesting to see what the sentiment of each song is, but it would be time-consuming to listen to every song and my interpretations of the songs could be biased. However, using a recurrent neural network (RNN) to analyze the lyrics of each song can provide some insight into the overall emotions of a song - happy, sad, relaxed, or angry. Using an RNN, I can also compare the network's output with my own interpretations of lyrics.

In order to perform sentiment analysis on songs based on their lyrics, an algorithm that is able to process text data efficiently and accurately is needed. Algorithms such as logistic regression and Naïve Bayes can be used to classify the text data by using vectorization methods such as bag-of-words or TFIDF. However, these methods can result in a loss of context from the original text, and create a less accurate model. Recurrent neural networks, however, are able to process text data better than other algorithms since they are able to update and maintain memory across different steps in the network. In RNNs, instead of processing information in a feed-forward manner like other neural networks, processes inputs in a loop, adding information from the past into its calculations for the present inputs. This mechanism makes RNNs well-suited for sequential data, such as text data and time series data. However, RNNs do have multiple weaknesses including exploding and vanishing gradients, and they have limited context as they only hold so much information from previous steps. To address some of these limitations,

Long-Term Short Memory (LSTM) will be used. LSTM contains a cell with input, output, and forget gates in order to control the information coming in and out of the cell. Information is assigned a value at these gates, and if the value is high enough, the respective action is taken - information written into the cell, discarded, or used as output from the cell. The data that will be used is song lyrics from almost 2000 songs across different genres, including rock, pop, and R&B. The target variable of this data is the mood, which includes happy, relaxed, angry, and sad.

Before the data could be used in the RNN, it had to be cleaned to remove punctuation and other extraneous characters, mostly with the use of regex. The cleaned lyrics were used to create a dictionary assigning each word to an integer based on how frequent it is, with 1 being the most common, and the lyrics were vectorized by transforming the words into the corresponding integers. Before going into the network, the lyrics were also all set to be a fixed input of 300 words by padding with zeros or by slicing the first 300 words. The network structure created was simple: an embedding layer, the LSTM layer, a fully connected layer, and a dropout layer for regularization purposes. The network was then put through the training loop based on the parameters defined, then evaluated using the test data.

After training and testing the network, my final model obtained an average test loss of 1.65 and a test accuracy of 41.3%. I also input some songs from my favorite artists, in which I determined the mood myself in order to compare the network's predictions to my interpretations, and the accuracy of this group of songs was 40%. This may be on the lower side, but it is based on how well the model matched my own interpretations, which can also be incorrect. Through this smaller dataset, I was also able to notice that the model is able to classify angry songs much more accurately than the others. These angrier songs are most likely easier to predict since they have many words with darker connotations, such as revenge and misery.

However, for the other moods, it can be more difficult to separate, especially happy and relaxed. With the simple vectorization method I used, only mapping words to integers, likely contributed to the network's difficulty in differentiating relaxing from happy or sad. Although the LSTM model is not perfect at classifying the sentiment of a song, only correctly identifying 4 out of every 10 songs, it can still classify a large volume of songs much faster than a person could. Despite the lower accuracy, the network is still able to give some knowledge on the frequency of moody words, as a song can have words with happier connotations but still be sad overall.

This project was very interesting, as I was able to combine my interest in natural language processing with my love for music to analyze song lyrics for information on their sentiments. The model created in this project is able to analyze the songs and provide output relatively quickly, although it is not as accurate as hoped. There are still many improvements that can be made to the model, but it can still be compared to how I and others interpret lyrics. While the network uses integers and calculations to analyze lyrics, people often connect lyrics to their own experiences or knowledge to provide their own interpretations. Neither approach can be seen as incorrect, and it is interesting to note the contrast in the two. In addition, I only focused on the analysis of the songs' lyrics in order to determine what feelings they provoked. In reality, songs convey emotions through both sound and lyrics, so sentiment analysis of songs only looks at one part of the song as a whole. Overall, it was fascinating to create and see the RNN in action, as well as get some insight on how a machine interprets song lyrics to decipher its sentiment.

References

Çano, Erion; Morisio, Maurizio. Music Mood Dataset Creation Based on Last.fm Tags. In: Fourth International Conference on Artificial Intelligence and Applications, AIAP 2017, Vienna Austria, 27-28 May 2017, pp. 15-26, DOI:10.5121/csit.2017.70603 (for dataset)