Title Page

Sentiment Analyzer

COMP 3106-A
FALL-2022

By:

Mitansh Desai	**Mihir Gupta**
101168117	**101172281**

Statement of contributions:

- Did each group member make significant contributions?
  Yes
- Did each group member make equal contributions?
  Yes
- Which aspects of the project did each group member contribute to?
  Mitansh primarily worked on implementing the code and helping with the method section for the report. Mihir took care of handling most of the report whilst helping contribute to the implementation.

Introduction

Sentiment analysis is the process of automatically identifying the sentiment of written text. It is a valuable tool in the modern world, where the amount of written communication continues to increase. In particular, the ability to analyze the sentiment of product reviews can be useful for companies looking to gauge the public's opinion of their products. In this report, we describe a sentiment analyzer that is designed to analyze the sentiment of product reviews.

Our motivation for solving this problem stems from the recent popularity of large language models such as GPT-3. This has sparked an interest in natural language processing (NLP) and the ability to create intelligent systems that can understand and generate human-like text. In this project, we aim to create a sentiment analyzer that can understand the sentiment of product reviews and provide useful insights to companies.

Related work in this area includes the use of machine learning algorithms and Neural Networks to classify text into different sentiment categories. These algorithms typically use a large dataset of labeled text to train a model that can then be applied to new, unseen text. However, our approach is unique in that we focus specifically on product reviews and use a specialized dataset to train our model.

This project’s significance lies in its ability to demonstrate a crucial way in which a vendor can analyze the sentiment regarding its product from outside sources, specifically 3rd party sources on the internet. By analyzing the sentiment of YouTube comments under a product's review or unboxing video, the vendor can gain valuable insights into the public's unbiased opinion of their product. The objective of this project is to develop a model that is trained on Amazon product reviews and can accurately analyze the sentiment of these YouTube comments (McAuley).

In this report, we will describe the problem of sentiment analysis and its relevance to product reviews. We will also describe our approach to solving this problem and present the results of our experiments. By the end of this report, we hope to have demonstrated the effectiveness of our sentiment analyzer and its potential applications in the real world.

Methods

Method 1:
The initial sentiment analyzer we implemented used a combination of natural language processing techniques and a temporal difference (TD) Q-learning algorithm to analyze the sentiment of product reviews. The TD Q-learning algorithm is a type of model-free reinforcement learning algorithm that is commonly used to solve problems involving sequential decision making. In this project, we use the TD Q-learning algorithm to learn the sentiment of product reviews based on the helpfulness ratings and star ratings provided by users.

To preprocess the text data, we remove punctuation, special characters, and stop words. We then perform stemming or lemmatization to reduce each word to its base form. This is done to make the text data more manageable and to improve the performance of the sentiment analyzer.

To evaluate the performance of our sentiment analyzer, we use a dataset of product reviews from Amazon. This dataset contains a large number of product reviews, along with helpfulness ratings and star ratings provided by users. We use this dataset to train our TD Q-learning model and to evaluate its performance on unseen data.

To evaluate the initial performance of our model, we conducted some qualitative testing by attempting to predict the sentiment of YouTube comments under a product review. However, we quickly realized that we were unable to achieve optimal results using Q-learning, a reinforcement learning algorithm that is **designed to learn from a sequence of actions and rewards in order to maximize a cumulative reward**. This is because the reviews are not a sequenced array of actions and rewards, making it **difficult to categorize the reviewText and overall star ratings as state-action pairs and assign an appropriate reward**. Despite our efforts to adjust hyperparameters such as **obtaining a dynamic learning rate and discount rate based on the helpfulness rating of a review and repetition of words**, we were unable to improve the model's performance. Hence, we pivoted our approach and used the alternative method described below.

Method 2:
We use a script for a sentiment analysis classifier that utilizes a neural network to predict whether customer reviews of a product are positive or negative. The script first imports the necessary libraries and modules, including pandas for data manipulation, nltk and tensorflow for natural language processing and machine learning, and matplotlib for data visualization.

Afterwards, a JSON file containing product reviews and metadata from Amazon, including 142.8 million reviews spanning May 1996 - July 2014 which includes details about the reviews (ratings, text, helpfulness votes, and more) which can be helpful to train sentiment on large sets of text (McAuley). The reviews are for the category of video games on amazon. This is an ideal dataset as it provides a large and diverse collection of largely unbiased text data and clear labels for the model to learn from. The text of the review is the main source of information for the sentiment analysis model and the overall rating of the product is a key indicator of the sentiment of the review and provides a clear label for the model to learn from. The model can use the overall rating to understand the sentiment of the review and learn to classify new reviews with similar sentiments.

To begin the process, we used pandas to read the JSON file and store it in a dataframe. This dataframe was then used to extract the review text, star ratings, and helpfulness ratings for each review. The star ratings were used to create training and testing labels for the machine learning model, with a value of 1 indicating a positive review and 0 indicating a negative review. In order to more efficiently and effectively model the data, we only included reviews with star ratings of 5 or 1, where a 5-star rating indicates a positive review and a 1-star rating indicates a negative review.

Next, the review text is cleaned by removing punctuation, special characters, and stop words, and by performing stemming or lemmatization to reduce each word to its base form. This cleaned review text is then tokenized, which means that each word is replaced with a unique integer. Finally, the tokenized reviews are padded with zeros to ensure that all the reviews have the same length (Menzli, 2022).

The machine learning model is a sequential model with four layers: an embedding layer, a global average pooling layer, a dense layer with a rectified linear unit (ReLU) activation function ($f(x) = \max(0, x)$), and a dense output layer with a sigmoid activation function (Sharma, 2022).

1. **Embedding:** This layer is used to represent words in a numerical format that can be used as input to a machine learning model. The input to the layer is an integer representing a word in the vocabulary, and the output is a dense vector of fixed size (the "embedding") that represents the word in a continuous space. The input_dim parameter specifies the size of the vocabulary, and the output_dim parameter specifies the size of the embedding vectors. The input_length parameter specifies the maximum length of the input sequences. This layer is vital because it allows the model to understand the meaning of the words in the product reviews and how they relate to one another (By: IBM Cloud Education).
2. **GlobalAveragePooling1D:** This layer is used to calculate the average value of the embedding vectors for each of the words in a sentence. It takes the average

of all the embeddings in the sequence and outputs a fixed-length vector. This layer is often used after an embedding layer to reduce the dimensionality of the input and make it more amenable to processing by a dense layer (By: IBM Cloud Education).

3. **Dense:** This is a fully-connected layer that takes an input tensor and applies a linear transformation to it. It has a number of units, specified by the units parameter, which determines the size of the output tensor. The activation parameter specifies the activation function to use, which determines the output of the layer given the input. In this case, the activation function is ReLU (rectified linear unit), which outputs the input if it is positive and 0 otherwise. This helps to introduce non-linearity to the model and improve its ability to learn complex patterns in the data (Sharma, 2022).
4. **Dense:** This is another fully-connected layer, similar to the previous one. In this case, the activation function is sigmoid, which maps the input to a value between 0 and 1. This is useful for binary classification tasks, where the output represents the probability that the input belongs to a particular class. In this case, the output layer would classify the sentiment of the product reviews as either positive or negative (Sharma, 2022).

The model is compiled with a binary cross-entropy loss function and the Adam optimizer, and the accuracy, precision, and recall are used as quantitative metrics to assess and validate the performance of the model. The model is then trained for 30 epochs, where one epoch is a complete pass through all the training dataset. Overall, this combination of layers is well-suited for sentiment analysis tasks and can effectively learn the patterns in the Amazon product reviews dataset in order to accurately classify the sentiment of YouTube comments.

Results

Qualitatively, the sentiment analyzer is able to predict the sentiment of individual reviews and provide a summary of the sentiment for a group of reviews. For example, the model can be given a list of reviews and it will output whether each review is positive or negative. Additionally, the model can provide a summary of the overall sentiment of the reviews, such as the average star rating out of 5.

Quantitatively, the performance of the model can be evaluated using various metrics such as accuracy, precision, and recall. These metrics measure the ability of the model to correctly classify reviews as positive or negative. For each epoch completed by the model, it prints out the loss, accuracy, precision, and recall for both the training and validation set. For example, for the 30th epoch in the results reported, the model had a binary cross entropy loss of 5.65%, an accuracy of 98.03% and a precision of 98.56% on the training set, and a loss of 16.19%, an accuracy of 95.02% and a precision of 95.64% on the validation set. The model also had a recall of almost 100% on both the training and validation sets, indicating that it was able to identify all of the positive reviews in the dataset. This suggests that the model is able to effectively capture the sentiment of the reviews and classify them accurately.

Overall, the results of this sentiment analyzer demonstrate its effectiveness in predicting the sentiment of customer reviews and providing a summary of the sentiment of a group of reviews. The high accuracy and precision scores achieved on both the training and validation sets suggest that the model is able to generalize well to new data and is reliable in its predictions. The high recall score further confirms the model's ability to identify positive reviews with a high degree of accuracy. These results demonstrate the potential of this sentiment analyzer to be used in a variety of applications where understanding the sentiment of customer reviews is important. However, when playing around with more qualitative testing, we have noticed that the model struggles to predict the sentiment of the review if it is very short in length and only consists of a small amount of meaningful words.

Discussion

The method used in the code is a machine learning approach to sentiment analysis of reviews. The method involves preprocessing the review text data by tokenizing and padding the reviews, and then training a neural network on the preprocessed data to predict the sentiment of the review as positive or negative (Menzli, 2022).

One potential implication of this approach is that it can provide accurate and efficient sentiment analysis of large amounts of review data. By training a neural network on the data, the model can learn to identify patterns and features in the text that are indicative of positive or negative sentiment, and make predictions based on those patterns. This can be more accurate and efficient than manual analysis of the text data.

However, there are also some limitations to this approach. One limitation is that the model's performance may be dependent on the quality and quantity of the training data. If the training data is not representative of the broader population of reviews, or if there is a limited amount of data available for training, the model's performance may suffer. In addition, the model's predictions may be influenced by any biases present in the training data.

Another potential limitation is that this approach may not take into account the context in which the review was written. For example, the model may not be able to differentiate between a review that is expressing frustration with a product and a review that is expressing excitement about a product, even though the words used in the review may be similar. As a result, the model's predictions may not always accurately reflect the sentiment of the review.

There are a few directions that the code could be improved upon in the future:

1. **Preprocessing:** The code could be modified to perform more advanced preprocessing techniques on the text data, such as removing numbers or handling emojis.
2. **Hyperparameter tuning:** The code could be modified to perform hyperparameter tuning, such as using grid search or random search, to find the optimal values for the model's hyperparameters.
3. **Evaluation metrics:** The code currently only uses accuracy as an evaluation metric, but other metrics such as a F1 score could be used to get a more comprehensive understanding of the model's performance.
4. **Data imbalance:** The code currently only uses a subset of the data for training and testing, and the proportion of positive and negative reviews in the dataset may not be balanced. Balancing the data using techniques such as oversampling or undersampling could improve the model's performance.

5. **Incorporating additional features**: The code could be modified to incorporate additional features from the data, such as the review's length or the product's price, to see if they improve the model's performance.
6. **Transfer learning:** The code could be modified to utilize pre-trained models such as BERT or GPT-2, and fine-tune them on the sentiment analysis task, to see if this improves the model's performance.

References

1. By: IBM Cloud Education. (n.d.). *What is supervised learning?* IBM. Retrieved December 16, 2022, from https://www.ibm.com/cloud/learn/supervised-learning
2. McAuley, J. (n.d.). *Amazon product data*. Amazon review data. Retrieved December 16, 2022, from http://jmcauley.ucsd.edu/data/amazon/links.html
3. Menzli, A. (2022, November 14). *Tokenization in NLP: Types, challenges, examples, tools*. neptune.ai. Retrieved December 16, 2022, from https://neptune.ai/blog/tokenization-in-nlp
4. Sharma, S. (2022, November 20). *Activation functions in neural networks*. Medium. Retrieved December 16, 2022, from https://towardsdatascience.com/activation-functions-neural-networks-1cbd9f8d91d6