Sentiment Analysis: Amazon Product Review Score Prediction on Video games and Musical Instruments by BERT and Bag of Words Model

Jing Zhou, Minjie Shen, Jiamu Li

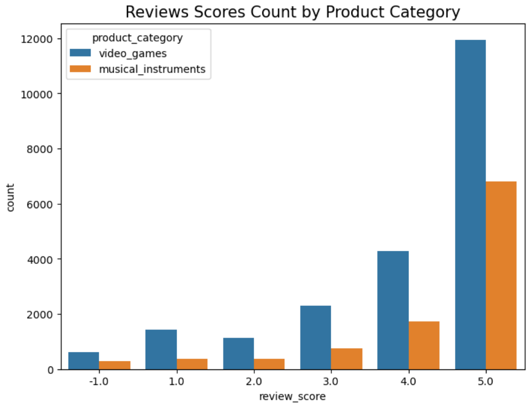
December 9, 2023

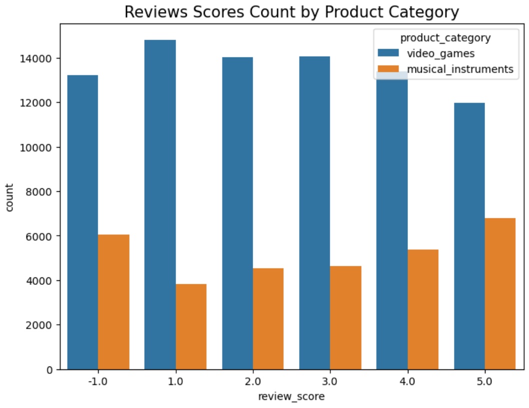
Abstract

Sentiment Analysis is a subfield of Natural Language Processing, which focuses on extracting and analyzing the information in sentences and predicting emotions. Sentiment Analysis has become one of the most popular topics in the world, and it has been used in many areas, such as shopping websites, customer services, and social media. In order to better understand how it works, we decided to implement it by ourselves. In this project, we used Amazon product reviews as the dataset to train two NLP models, BERT and Bag of Words, eventually predicting scores of new reviews.

Data Processing

To train the model, we have to preprocess the dataset. We use `WordNetLemmatizer`, `stopwords`, `word_tokenize` from `nltk` library to preprocess the data. We first lowercase the review and remove all non-alphanumeric characters, punctuations, and links. Then we tokenize the review and remove stopwords. We also perform lemmatization on tokens. After cleaning reviews, we upsample the minority in the data since we found that the number of score 5.0 is much larger than that of other scores. Thus, we use `resample` from `sklearn` library to upsample the number of other scores to the same number as score 5. To feed the data into the model, we also use `CounterVectorizer` from `sklearn` to transform text data into numeric data. Finally, we use `train_test_split` from `sklearn` to split data randomly into training set and test set.





Models

Bag of Words

Our bag of words model is implemented using neural network. The model takes a vectorized text as input and outputs a predicted score. It has one hidden layer with 128 neurons followed by the ReLU activation function. To ensure that outputs fall into a range from 0 to 5 (here 0 represents -1 score), the model applies the sigmoid function on the output and multiplies it by 5. We use MSELoss as the criterion and Adam as the optimizer. We train the model with 512 batch size for 20 epochs to obtain a good result.

BERT

We use `BertForSequenceClassification`'s pretrained model `bert-base-uncased` from `transformers` library to train our data. This model is pretrained with masked language modeling and next sentence prediction. To train the model, we have to add attention masks and segments to the data. We use `BertTokenizer` tokenizer of `bert-base-uncased` model from `transformers` with 64 max sequence length to generate attention masks and segments. The criterion is CrossEntropyLoss and the optimizer is AdamW. We train the model with 128 batch size and 10 epochs to obtain a good result.

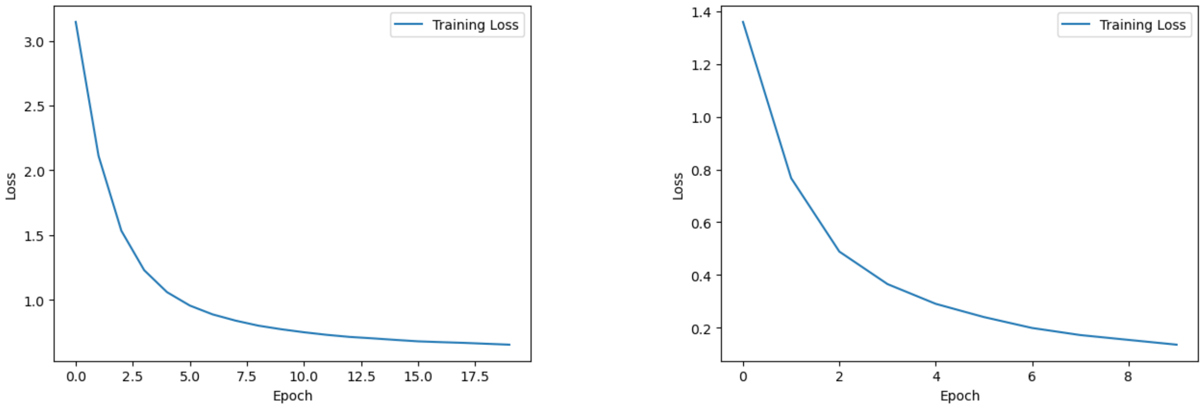
Evaluation

Methodology

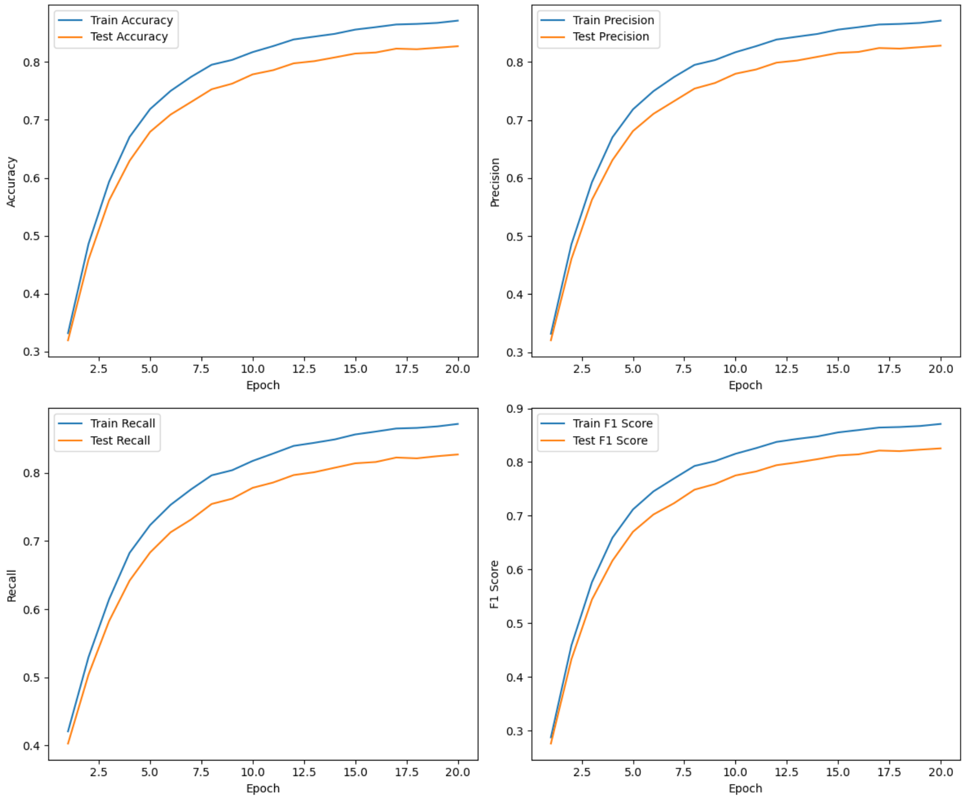
We do the same evaluation on two models. For each model, we record the average loss on training set during each epoch. We also record the predicted score of both training and test set to obtain accuracy, precision, recall, and f1 score. By checking these, we can have a brief view on how these model perform.

Results

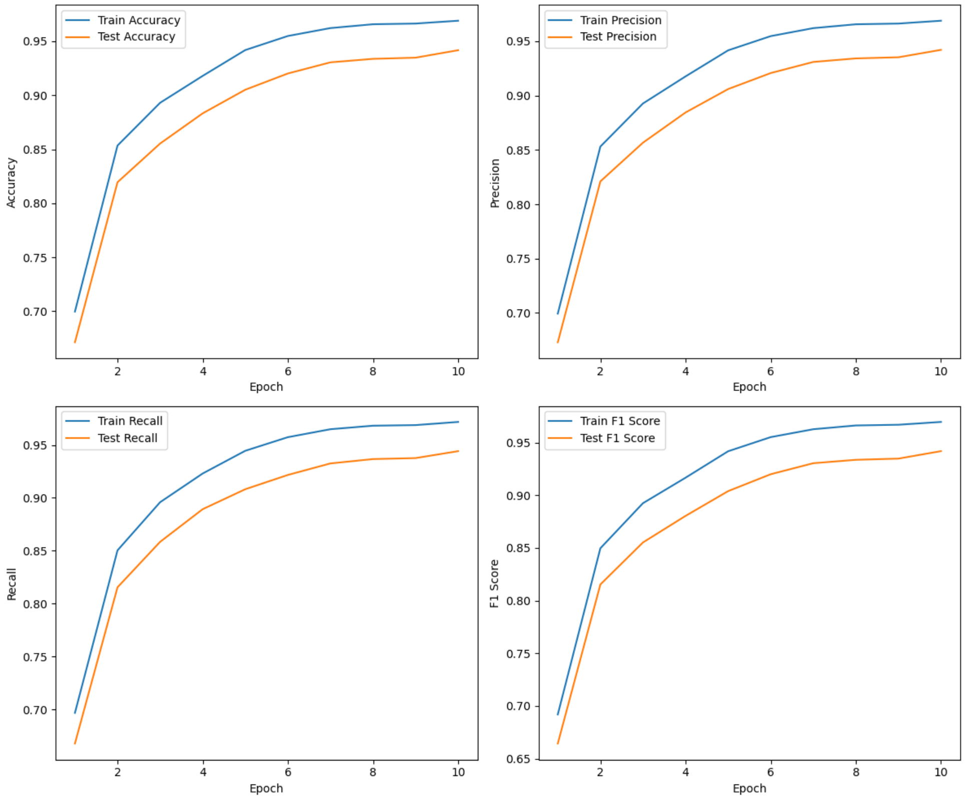
Loss, Accuracy, Precision, Recall, F1 Score



BOW(left) vs. BERT(right)


Train Accuracy
Test Accuracy
Accuracy
0.3
0.4
0.5
0.6
0.7
0.8
2.5
5.0
7.5
10.0
12.5
15.0
17.5
20.0
Epoch
Train Precision
Test Precision
Precision
Train Recall
Test Recall
Recall
Train F1 Score
Test F1 Score
F1 Score
0.9

BOW


Train Accuracy
Test Accuracy
Accuracy
0.65
0.70
0.75
0.80
0.85
0.90
0.95
2
4
6
8
10
Epoch
Train Precision
Test Precision
Precision
Train Recall
Test Recall
Recall
Train F1 Score
Test F1 Score
F1 Score

BERT

Conclusion

By comparing the results of BOW and BERT model, it is obvious that BERT is on a higher level. BOW model has 0.65 loss at the end of the training whereas BERT model has only 0.13 loss. In addition, BOW model has 0.82 on accuracy, precision, recall, and f1 score but BERT has 0.94 on those. We also test them with new reviews. For a positive review ‘This item is perfect’, BOW model gives score 4, but BERT model gives score 5. For a negative review ‘This item is awful’, both model give score 1. Generally speaking, we always want to give score 5 when we say a product is perfect. Thus, BERT has a better performance.

Future Work

We would like to adjust BOW model, such as adding hidden layers and fine-tuned parameters, to see if basic neural networks can have the same performance as BERT model. We can also try other models such as Long Short-Term Memory network or Generative Pretrained Transformer and compare these models to find out which model performs the best.

Work Cited

Dataset

https://www.kaggle.com/datasets/mohammadbilalgul/amazon-review-data

GitHub Repository

https://github.com/Jzhou271/npl_final_project_amazon_review