Amazon Product Rate Predictor

JeongYoon Lee
jeongyoon.lee.1
@stonybrook.edu

Dovid Scheinberg
dovid.scheinberg
@stonybrook.edu

Andrew Chu
andrew.chu.1
@stonybrook.edu

Abstract

As e-commerce continues to become a more widely used means of purchasing goods, the more vital it is for consumers to be able to accurately determine if the product is the right fit for them. The number of stars a product has is often a major deciding factor for this, so they must be as reliable and consistent as possible. We propose a way to evaluate a product based solely on its text reviews: by training a neural network to assign a score to a review. Affordability and accessibility for these tasks is also important so in order to preserve that we fine-tuned a pretrained deep neural network to learn in a short amount of time.

1 Introduction

In this project we hope to train a neural network to predict the scores of amazon reviews. This could be useful in identifying outlier scores or for rating reviews with no score assigned. Often a review's score does not match the sentiment of the review itself. It could be possible to allow users to write reviews with no score and then evaluate the scores afterward to provide more accurate, less biased scores for products.

Shrestha and Nasoz converted data to vectors using paragraph vector trained a GRU on the resulting data. They then built a proof of concept web application that notifies the reviewer if their manually assigned score is semantically inconsistent with their text review.[1] Shah et. al used POS tagging, bag of words, and TF-IDF in conjunction with KNN, SVM, Decision Tree, and other non-neural network classifiers to classify text reviews into negative or positive sentiments.[2] Xie et. al used BERT-Large along with several complicated noising algorithms and data augmentation techniques such as back-translation for semi-supervised learning to achieve an accuracy of 65% on 5 class amazon prediction.[3]

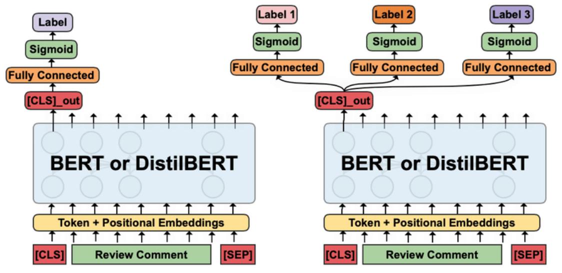
[1]https://arxiv.org/ftp/arxiv/papers/1904/1904.04096.pdf
[2]https://ieeexplore.ieee.org/abstract/document/8376299

2 Your Task

The input of this task is an amazon text review, and the output is a score ranging from 1 through 5 where 1 represents a more negative sentiment and 5 represents a more positive one. Our goal is to achieve an accuracy of 60% with our model using DistillBERT.

3 Model 1 (2-4 paragraphs)

HuggingFace's DistilBERT model was used as a base for our training and fine-tuning. It is considered as a "smaller, faster, cheaper, and lighter" version of the BERT model we discussed in class so it can take a reasonable amount of time to train. It is widely used in the field of Natural Language Processing. More specifically, the DistilBERT model is a distilled version of the BERT model, which means it approximates it while using 40% less parameters, running 50% faster, and preserving over 95% of the performance[4]. The original BERT model was trained on an extremely large corpus of Wikipedia articles.



[3]https://arxiv.org/pdf/1904.12848v6.pdf
[4]https://huggingface.co/docs/transformers/model$_{doc/distilbert}$

4 Experimental Setup

4.1 Dataset Details

The amazon_reviews_multi (English) dataset was used. This was perfect for us because our project was created in hopes to predict the score of a text review, and this dataset gives us 200,000 English text amazon reviews and their titles to train on and their corresponding stars, along with some other information that we did not use such as category. This dataset indeed has a balanced distribution of scores (20% of each star from 1-5).

4.2 Evaluation Measures

For regression, we used mean squared error, mean absolute error, r2 score, and accuracy after rounding predictions to the nearest label. For classification, we used a simple accuracy and loss score.

4.3 Model Implementation Details

For our model we first loaded the DistilBERT model from HuggingFace and froze the embeddings and the first 4 of 6 layers. We also loaded the amazon_reviews_multi (English) dataset, where each review was a JSON instance consisting of a review_id, product_id, reviewer_id, stars, review_body, review_title, language, and product_category. To get our training data, we paired each of the tokenized training data reviews with its corresponding label (score out of 5). We did the same for the validation data. For the classification version we set the number of labels to 5, but for the regression model we set the number of labels to 1 which causes the model to output regression results.

4.4 Implementation Details

We heavily utilized HuggingFace's transformers library so we can use their pretrained DistilBERT-uncased model and tokenizer. We also used their datasets library to conveniently access their amazon_reviews_multi (English) dataset. Pytorch was used for training. For both regression and classification, we trained on a batch size of 16 and a dataset size of 20000. And then we divided each label value by 5 to make it a label value between 0 and 1.

5 Results

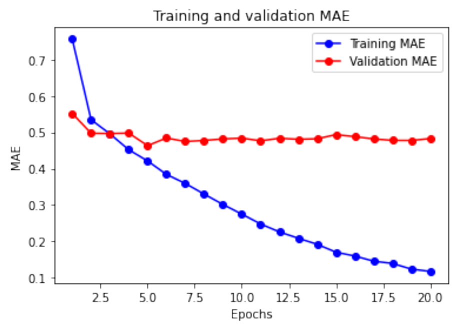
5.1 Regression Analysis

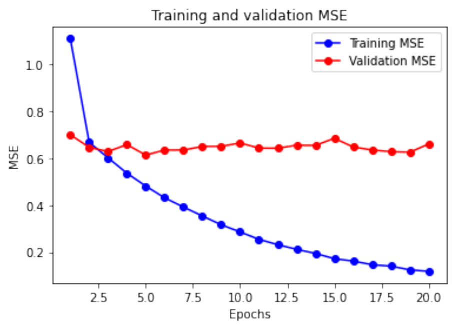
Here you can find the results of training the DistilBERT model on the amazon_review_multi dataset. We used different metrics for regression and classification, but measured accuracy and loss for both.

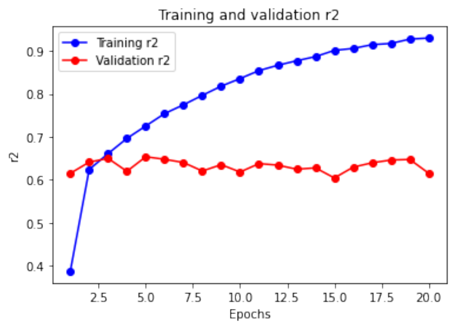
#	Epochs	MAE	MSE	r2	Accuracy	Loss
1	3	0.5124	0.6749	0.6136	0.5601	0.0244
2	5	0.4633	0.6130	0.6536	0.6006	0.0221
3	20	0.4828	0.6601	0.6151	0.5938	0.0238

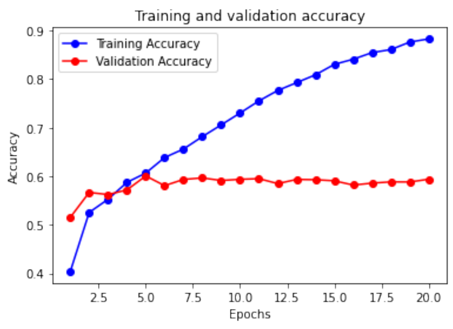
Table 1: Regression results

(batch size: 16, training samples: 20k, learning rate : 0.00003)



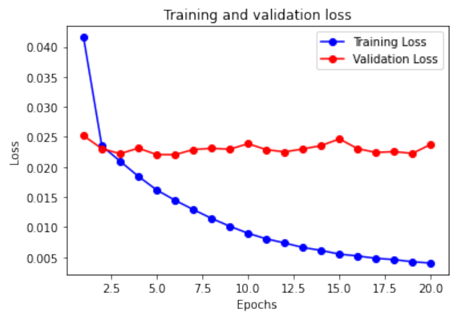






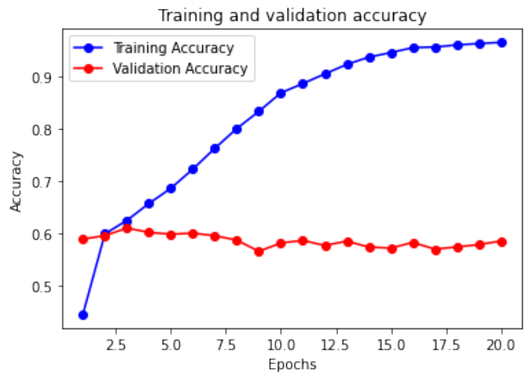
Model	Epochs	Accuracy	Loss
1	3	0.6096	0.3068
2	5	0.5978	0.3203
3	20	0.5847	0.7413

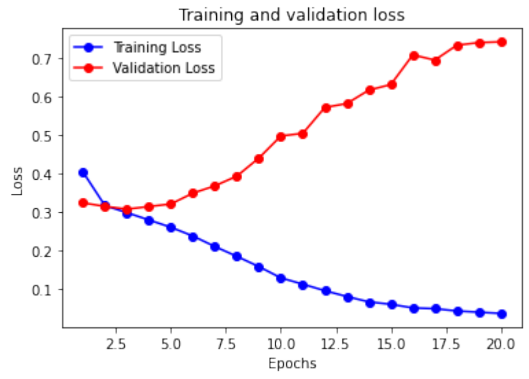
Table 2: Classification Result



5.2 Classification Analysis

(batch size: 16, training samples: 20k, learning rate : 0.00003)





It appears both models performed very similarly. Both models began to overfit, though the classification model seemed to overfit sightly worse than the regression model. Both achieved a peak accuracy of around .6, .5 lower than the highest score we could find for 5 label classification for Amazon reviews.

6 Code

https://drive.google.com/drive/folders/1TBdh73g VkFKE98oZdPCWSle3wn7o_reS?usp=sharing

Using GPU from Colab (CUDA)

7 Conclusions

We used the DistilBERT model to predict the product score based on review data from Amazon. The DistilBERT model is a general-purpose, pre-trained version of BERT that is 40% smaller, 60% faster than BERT, and keeps the language understanding capabilities. So we are trying to use a pre-trained model from HuggingFace's transformers library. At first, we tried to freeze layers in various ways, but since this is the DistilBERT model, it doesn't seem to make much difference. So we decide to freeze the first 4 layers and set the learning rate to 0.00003 and batch size to 16 across several epochs. If it trained more than 3 to 5 epochs, it will have some overfitting issues and not much difference between regression and classification models. We got r-squared values over 0.6 for the evaluation measures, and our model seems to be well explained by our dataset. Also, both models have close to 60% accuracy, we can say that this model is good at predicting product scores. We will use a model other than DistillBERT in a future study to compare its performance with this model.

8 References

- Xie, Q., Dai, Z., Hovy, E., Luong, M.-T., Le, Q. V. (2019). Unsupervised data augmentation for consistency training. https://doi.org/10.48550/ARXIV.1904.12848

- Shrestha, N., Nasoz, F. (2019a). Deep learning sentiment analysis of amazon. Com reviews and ratings. https://doi.org/10.48550/ARXIV.1904.04096

- Haque, T. U., Saber, N. N., Shah, F. M. (2018). Sentiment analysis on large scale Amazon product reviews. 2018 IEEE International Conference on Innovative Research and Development (ICIRD), 1–6. https://doi.org/10.1109/ICIRD.2018.8376299