CS371-2025S: Final Report
“Zero-shot Rating Regressor Based on BERT”

Seungjun Kim

1. Introduction

A lot of LLM models such as BERT shows high indexes in finding inherent meaning in sentence. Because of its general-purpose, combined with another architecture such as regressor, BERT-based models can do tasks such as rating score prediction by review text. However, review texts should be considered with the category the review appointing to. For example, ”So terrifying” is a negative review text for a lot of category, whereas can be a positive review for a category like horror movie. I was motivated to regressor model to map review text to rating score between 1 and 10 considering category reviewed. Now the problem is that given review text and random category, map rating score from 1 to 10. The main contributions of this paper are: (1) Different from such rating prediction tasks, score prediction is based on the given category. (2) New regressor deep learning architecture using multi-input text embedding. (3) For any unseen class, the rating regressor can give a reasonable score, not only for the goods reviewed. Review and discussion can be evaluated in any category. News, any surveys or any evaluations are accompanied with some review. But analyzing reviews one by one is impossible. Rating is best metric for review. For example, movie star score shows average evaluation of audience. If rating regression is used in a policy and laws with enormous public reviews from survey or interviews, states can propose policies or laws that focus more on the opinions of the people. In this way, we can contributes to Elimination of Inequality and Global Harmony.

2. Related work

There are papers[(Ramya et al., 2022), (Ali et al., 2022), (de Paula e Santiago, 2023)] of task-related AI projects that predict stars or sentiments together. However, these projects were predicted by classification rather than regression. However, evaluation using continuous scores is more accurate and easier to compare than classification of about 5 classes. Additionally, There is a problem that other contextual features are not reflected.

3. Method

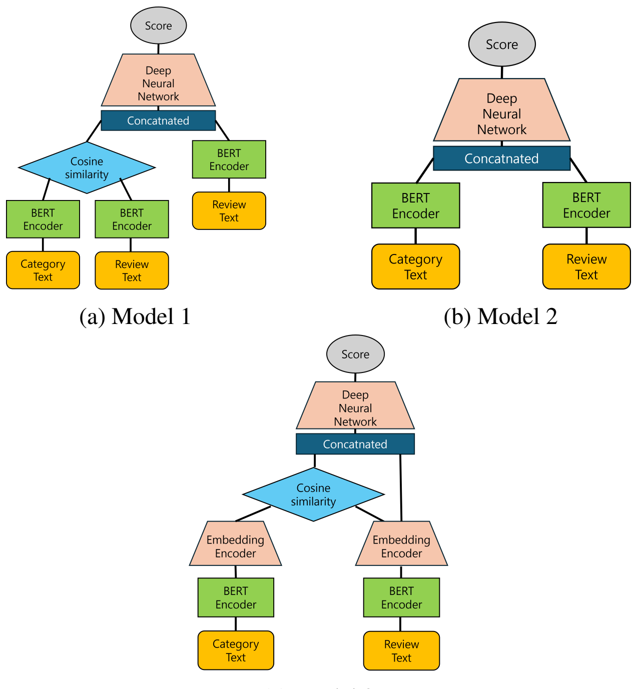
In base, without fine-tuning, I improved performance by neural network-based regressor combined with similarity metric. I modeled 3 different architectures. They are different in that how category vector is added to review text vector feature. The figures[1] refer to the model’s architectures. Each models’ Regressors have neural network layers with ReLU and dropout layer. The size is different because of different input size. First model[1a,1] concatenate cosine similarity between category embedding and review embedding with expanded dimension. BERT embeddings of contextually similar texts tend to have similar direction. By doing so, similarity feature is added to review text. Because cosine similarity is in the range from -1 to 1, I increased the scale by 5 and concatenate with expanded to 32 dimension for the feature not to vanish. Second model[1b,2] put the category’s feature on the same line with review’s feature. Third model[1c,3] has similar architecture with CLIP, zero-shot learning model. review and category have independent encoders. Each encoded vectors’ similarity is concatenated to encoded review text vector. Training[4] section used optimizer Adam with learning late 0.001, epoch 1. Loss function is MSE(Mean Squared Error).

Algorithm 1 Forward of Model1

```
**Input:** tokenized review $r_i$, tokenized category $c_i$
**Output:** score
sim=CosineSimilarity($r_i$,$c_i$)
sim=sim.expand(32,-1)
convec=concat(sim,$r_i$)
score=DNNRegressor(convec)
**return** score
```

Algorithm 2 Forward of Model2

```
**Input:** tokenized review $r_i$, tokenized category $c_i$
**Output:** score
convec=concat($c_i$,$r_i$)
score=DNNRegressor(convec)
**return** score
```



(a) Model 1 (b) Model 2 (c) Model 3

Figure 1. Comparison of the three model architectures.

Pre-trained LLM model BERT was used for the foundation model. BERT makes text tokens to word embedding with contextual meaning. Due to lack of computing resources, fine-tuning is omitted. Uncased base BERT is for English and yield 768-dimension embedding vectors for all tokens. The first [CLS] token contains overall contextual meaning embedding, so I used only first embedding.

Algorithm 3 Forward of Model3

```
**Input:** tokenized review $r_i$, tokenized category $c_i$
**Output:** score
revvec=NNencoder1($r_1$)
catvec=NNencoder2($c_1$)
sim=CosineSimilarity(revvec,catvec)
sim=sim.expand(32,-1)
convec=concat(sim,revvec)
score=DNNRegressor(convec)
**return** score
```

Algorithm 4 Training

```
**Input:** review datas $R$, category datas $C$, score labels $L$
**for** $i = 1$ **to** len($R$) **do**
    outputs = model($R_i$,$C_i$)
    loss = MSE(outputs, $L_i$)
    zerograd
    backward
**end for**
```

4. Experiments

I trained and evaluated all models with concatenated kaggle review datasets. The category contains many kinds of goods or services. Data were preprocessed to ensure that as many category samples remained as possible. Ratings are scaled to max value 10. Learning and testing were performed with each evenly distributed rating data[2]. All learning and evaluation was performed with A100 GPU and high-capacity RAM in the Colab Pro environment with pytorch, while other activities such as data processing were performed with CPU in the Colab Pro environment. Making different 3 regressor classes, I trained them with same dataset on train and test, then conducted quantitative and qualitative evaluations. Qualitatively, all 3 models do well in grading solid sentimental reviews. Sentimentally negative review scores 2 3, whereas sentimentally positive review scores 8 10. However, for same review text, different categories have a very small effect on the score, ranging from 0 to 0.1. Simply put, the impact of categories on scores is too minimal compared to the review text. Quantitatively, MSE represents average rating score error. The value range from 1.26 to 1.34, reasonable scale considering test rating distribution is very discrete. The evaluation results for various regression model evaluation indicators are shown in the following table[1]. The prediction error is similar on 3 models, but model 1 with cosine similarity as feature performs better than the number of parameters. As the distribution of dataset[2] shows that rating distributed is not continuous, making it hard to map score correctly. As a result, average error exceeds 1. All models show small errors in predicting evaluation scores, but learning zero-shot regression by adding an unprecedented category feature that was

Table 1. The evaluation results for various regression models.

MODELS	MODEL1	MODEL2	MODEL3
# OF PARAMETERS	222081	1395329	435969
MAE	1.2772	1.2552	1.3386
MSE	2.9416	2.9345	3.2044
RMSE	1.7151	1.7130	1.7901
R2	0.6287	0.6295	0.5955
MAPE	31.3065	30.1162	32.4278

originally expected from the main attribute was not done well. I expect three major causes for this:(1) Unbalanced categories. Plot[3] shows cumulative percentage in order of most frequent category. One category accounts for half of the total data, and the proportion of the top 50 categories already exceeds 95% of the total data. For this reason, it is impossible to learn a neural network tailored to category vectors in various directions that have not been seen. (2) Lack of fine-tuning. Not fine-tuned, BERT can remove features from text needed for review task. (3) Limitations of Deep Neural Networks. Depending on the data, there are reviews in which categories have a great influence, and there are reviews with little influence. Adding a similarity or category vector to a neural network as it is can create a biased neural network in which the weight of the category vector or similarity hardly affects the result.

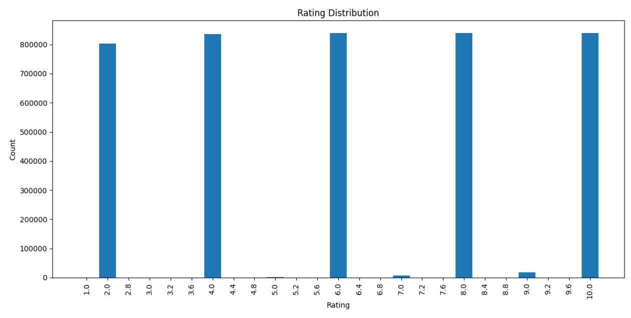


Figure 2. Rating Distribution

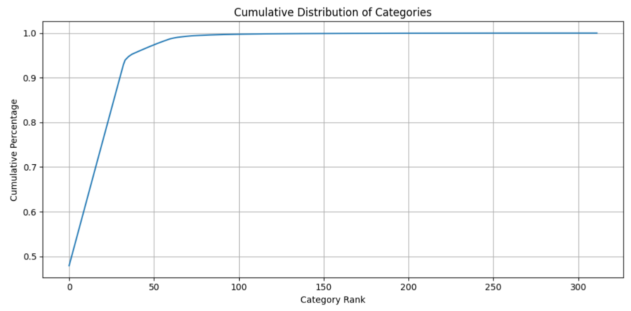


Figure 3. Cumulative Category Distribution

Table 2. Number of Unique Categories on each Rating.

RATING SCORES	# OF UNIQUE CATEGORIES	
ALL	312 2.0	237
4.0	223	
6.0	240	
8.0	266	
10.0	303	

5. Conclusion

The problem was defined, and three different zero-shot review register deep learning models were implemented with the Pytorch library in Colab Pro's GPU environment. Although the scores of emotionally positive or negative reviews were well predicted, there was a problem that the difference in the result values was too small when different categories were input for the same review.The distribution of review categories was too biased, so category-based zero-shot learning was not good. After that, we intend to learn based on unbiased categories while increasing the data further. In addition, rather than combining independent BERT and Reressor, it is expected that some layers of BERT can be fine-tuned or trained from scratch using a transformer model to better reflect the category embedding. By training three models while resources were scarce, the learning range of the model was too small and the epoch was very small. In a more and better resource environment, you can achieve better results by increasing the size of the model. In addition, when I originally proposed the project, I tried to create a Korean-based model with Korean data, but I gave up because the Korean data was too insufficient. If I get a Korean data source, I would like to experiment with BERT's Korean-based fine tuning version of the koBERT model. The table[2] shows the number of category types according to each rating score. The lower the score, the more absolutely the category is insufficient. In other words, it is difficult to classify low-scoring reviews by category. Also, overall, there are too few 312 categories for about 4 million data. It is a problem of data collection, but it is also a problem of small dataset types. If you learn similarity with the 768-dimensional embedding vector and learn only 312 directions, of course, the performance of the model decreases significantly. I think that various datasets are essential for the development of more diverse artificial intelligence models.

References

Ali, A., Qamar, U., et al. Effectiveness of fine-tuned bert model in classification of helpful and unhelpful reviews. *Comput Intell Neurosci*, 2022:9051848, 2022. doi: 10.1155/2022/9051848. URL `https://pmc.ncbi.nlm.nih.gov/articles/PMC9051848/`.

de Paula e Santiago, C. M. F. Rating prediction of product reviews: An approach based on the bert language model. Master's thesis, Universidade Federal do Rio Grande do Sul, 2023. URL `https://lume.ufrgs.br/bitstream/handle/10183/257982/001168342.pdf?sequence=1`.

Ramya, S., Radhika, R., and Vijayalakshmi, S. Bert over linear regression algorithm. *Journal of Pharmaceutical Negative Results*, 13(S04):30–38, 2022. doi: 10.47750/pnr.2022.13.S04.004. URL `https://www.pnrjournal.com/index.php/home/article/view/882`.