Reddit Comment Popularity

Billy Shi
SFU
lsa84@sfu.ca

Zachary Fong
SFU
zachf@sfu.ca

Abstract

Our project explores the application of DistilBERT integrated with various regression heads to predict the popularity of comments within the AskReddit subreddit on Reddit. This leverages natural language processing (NLP) and machine learning techniques, we aim to understand the effectiveness of a distilled transformer model in analyzing textual content for sentiment and popularity. The project tests different regression head architectures—ranging from simple linear layers to more complex Multi-Layer Perceptrons (MLP) and also a LSTM networks to explore the most effective approach in predicting comment scores. By normalizing scores into a relative ratio against the highest scoring comment in the same thread, our models focus on understanding the relative engagement rather than absolute values, which varies widely across posts. The study conducts both within-domain and out-of-domain evaluations to assess generalizability and robustness of the models across different subsets of data.

1 Introduction

1.1 Background and Motivation

Social media platforms thrive on user interactions and engaging comments can play a crucial role in a positive user experience. However, the predictability of comment popularity is influenced by various factors, including content, length, and user engagement. This project explores the AskReddit subreddit, where users can post questions (posts) and answer with responses (comments), and investigates whether machine learning models can effectively predict comment scores. The primary motivation is to understand how a model consisting of DistilBERT (Sanh et al., 2019) uncased with a regression head performs when predicting comment popularity and to improve the predictive capabilities of the model in this context. We will be evaluating different regression head architectures and comparing them against a baseline fully connected linear layer regression head.

1.2 Task Definition

In our project, we will use a dataset consisting of the top 1000 posts along with their corresponding comments for both training and within-domain evaluation purposes. Additionally, for out-of-domain evaluation, we will utilize posts ranked from 1000 to 1100 inclusive, along with their associated comments. Each post includes a collection of comments which have an assigned score indicating its popularity and user engagement level. This score is determined by user interactions such as upvoting (increasing the score by 1) or downvoting (decreasing the score by 1). Instead of directly using these raw scores, we aim to predict a ratio that reflects the relative popularity of each comment within its respective post. This ratio is computed by dividing the score of each comment by the highest scoring comment within the same post. By normalizing the scores in this manner, we ensure that our model focuses on capturing the relative popularity of comments within each post, rather than absolute score values. This approach enables us to develop a metric that better reflects the comparative engagement levels of comments within the context of their corresponding posts.

The input into the model for a given comment will then be the post title of the comment and the comment concatenated together, then tokenized using the distilbert uncased tokenizer. Consider table 1 for comments of the same post. For the first row, the input into the model would be tokenize("What's your favourite colour? Blue"), the ground truth would be 1, and the goal of the model is to predict close to the ground truth.

1.3 Objective and hypothesis

The goal of this project is to assess the effectiveness of using DistilBERT combined with a regression

Title	Comment	Comment score	Highest comment score for the post	Comment score ratio
What's your favourite colour?	Blue	100	100	100/100 = 1
What's your favourite colour?	I dont have one	3	100	3/100 = 0.03
What's your favourite colour?	Sometimes it changes	35	100	35/100 = 0.35

Table 1: Input example

head to predict Reddit comment popularity. The project explores both within-domain and out-of-domain evaluations across different inputs to gauge the model's performance under inputs with different characteristics.

We hypothesize that the MLP regression head will outperform all of our other regression heads. We believe that our models may bias towards predicting comments as unpopular because unpopular comments are much more frequent in our training dataset. We also predict that the models will predict long comments as accurately as short comments and that the models will perform better when evaluated using within domain data when compared to out of domain data.

2 Related Work

2.1 Fine-tuning BERT for a regression task: is a description enough to predict a property's list price?

We referred to a medium article (Galtier, 2021). This article demonstrates a project by training a model to take a housing description in French and predicts its housing price. This model used is CamemBERT (Martin et al., 2020) for tokenizing and computing embeddings which is a RoBERTa (Liu et al., 2019) architecture pretrained on the French sub-corpus of OSCAR. From the output of the CamemBERT, which is a dimension of 768 embeddings, it attaches a dense linear layer with drop out as the final regression head for predicting housing price. The training process uses MSE as loss function, and both fine tuning on the entire BERT and the regression head. After 5 epochs the training loss has shown significant decrease from 0.33 to 0.06, for test loss shows a more steady decline with 0.24 to 0.17. The author compare the performance with his previous work with tabular set of numerical and categorical features as input to LSTM model to predict list prices. Which has shown similar result to BERT based model with listing description as input. This task is similar to our objective, a regression task base on BERT model with text as input.

2.2 Enhancing Automated Essay Scoring Performance via Fine-tuning Pre-trained Language Models with Combination of Regression and Ranking

Another paper (Yang et al., 2020) we have looked at is to enhance Automated Essay Scoring by fine tuning LLM with regression and ranking. They introduced a structure of BERT model attached with a combination of a Regression head and a Ranking head. Which they summarized that "Regression loss and ranking loss are two complementary losses." This could be an addition point to try on our model by ranking the comments as an extra output of the model, and combine those two outputs.

3 Approach

3.1 Input

The input into our model consists of the combined title text and the comment text separated with a [SEP] token, then passed into a tokenizer. The [SEP] token is used to delineate between the title and the comment which allows the DistilBERT's attention mechanism to attend to the title and comment separately. This could potentially allow our model to learn the relationship between the title and comment, and improve predictions. The tokenizer we use is the distilbert uncased tokenizer from Hugging Face with truncation set to true, padding set to true, and the max length set to 512 tokens. We originally passed in these tokenized inputs to our model in batches of 64, but found training times to be extremely long at over 1 hour per epoch.

To streamline the training process, given that only the regression head was subject to training (with DistilBERT's parameters being frozen), we pre-processed the inputs through DistilBERT to obtain embeddings for each input. We then were able to directly feed our regression head these inputs and reduce the training time to around 1 minute per epoch.

Along with the tokenized input, the comments associated score ratio was used as the ground truth to compare against the models predictions.

3.2 Output

As described previously, the output of our model for each input is the comment's score ratio. This score ratio is calculated by dividing the score of

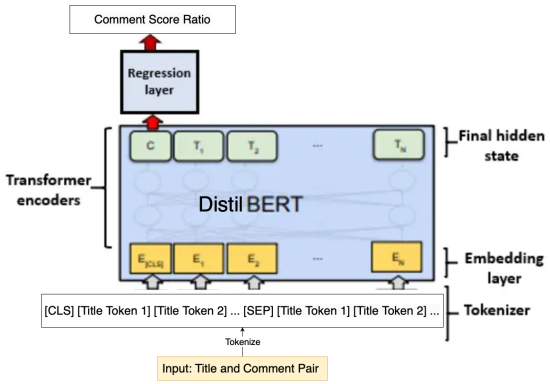

Figure 1: Our model's strucutre

a specific comment by the score of the highest-scoring comment within the same post. This metric serves as a normalized indicator of a comment's popularity relative to the most engaging content in the post, providing a standardized measure of success across different posts. During training, the model's objective is to minimize the discrepancy between its predicted score ratios and the actual score ratios derived from the training data.

3.3 Model

Our model architecture is similar to the medium article [Figure1], however we use English title and comment pairs as input. We have also decided to freeze the DistilBERT model and append an unfrozen regression head to DistilBERT's final embeddings layer. Our project explores three distinct regression head configurations to process DistilBERT's 768-dimension output embeddings:

- Single Linear Layer Regression Head (baseline model): This simple configuration channels the 768-dimensional embedding through a single linear layer, leading to a singular output neuron. This setup aims to directly map the complex embeddings to a predicted score ratio, offering a straightforward approach to regression.

- Multi-Layer Perceptron (MLP) Regression Head: This more complex configuration involves an MLP design, composed of four layers. The first three layers are equipped with ReLU (Rectified Linear Unit) activation functions and a dropout mechanism set at a rate of 0.1. The ReLU activations introduce non-linearity which enables the model to grasp more intricate patterns within the data. The dropout layers serve to mitigate overfitting by randomly deactivating a subset of neurons during the training phase, helping the model better generalize unseen data. The MLP ends with a linear output layer with a single neuron, similar to the simpler model. The layers of this MLP consist of 512, 256, 128, and 1 neuron(s) respectively.

- Multi-Layer LSTM : This architecture does not use the pre-computed embeddings from DistilBERT and instead captures its own context from the tokenized input. This model consists of 3 LSTM layers that each have 256 neurons, 0.15 dropout, bidirectionality, and a linear output layer for regression. This model functions as our own lightweight version of DistilBERT that we are able to fully customize and train on our dataset.

Both the multi-layer LSTM and the MLP's specific architecture (number of layers, layer sizes, and dropout rate) were refined through experimentation.

4 Data

4.1 Background info

The dataset for this project is derived from the Reddit corpus hosted on the SFU compute cluster. Reddit's structure is composed of various subreddits, each characterized by unique posting guidelines and thematic focus. We selected the AskReddit subreddit for our analysis due to its straightforward format: users post questions as titles, and the community responds in the comments section. This choice strategically limits our dataset to textual content, avoiding the complexity of analyzing multimedia elements such as photos, videos, or hyperlinks, which could potentially bias comment scores.

4.2 Train test split

After narrowing down the Reddit corpus to the AskReddit subreddit, we settled on using the top 1100 posts and their associated comments for training and testing.

Training dataset: We utilized 90% of the top 1,000 posts and their comments for training the model. This segment enables the model to learn from a rich variety of question and answer pairs, ensuring a broad representation of the AskReddit community's interaction dynamics.

Within domain evaluation: For testing the model's ability to generalize to unseen data while remaining within the familiar domain, we allocated the remaining 10% of the top 1,000 posts and their comments. This subset tests the model's predictive accuracy on new comments from previously encountered posts.

Out of domain evaluation: To assess the model's performance on completely new content, we used posts ranked from 1,001 to 1,100 and their comments. This evaluation measures how well the model adapts to predicting comment scores without prior exposure to the post and discussion threads.

4.3 Refining Dataset

During data analysis, we noticed an interesting characteristic of our dataset. The score and score ratio of our data was extremely right skewed. This meant that unpopular comments outweighed popular comments by a huge proportion. We worried that this may impact the performance of the models so we implemented a score frequency cap at 1e3 comments per score (not score ratio). This resulted in a 90% decrease in our dataset, with only scores of 7 and below being removed as seen in figure2. We now had two datasets: the full dataset and the refined dataset.

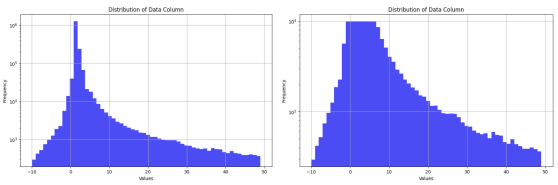

Figure 2: Log Score Distribution

5 Experiments

For our experiments, we will train and compare different linear, MLP, and LSTM regression models. We will first train the linear and MLP regression heads on our full dataset, then retrain them on the refined dataset to try and reduce low score ratio predictions. We will then train the LSTM regression model on the refined dataset to see if it can outperform our other models. We will be comparing and evaluating these models using MSE, RMSE, MAE, and $R\hat{2}$. The testing data (within domain and out of domain data) will also be refined if the model is trained on the refined training dataset.

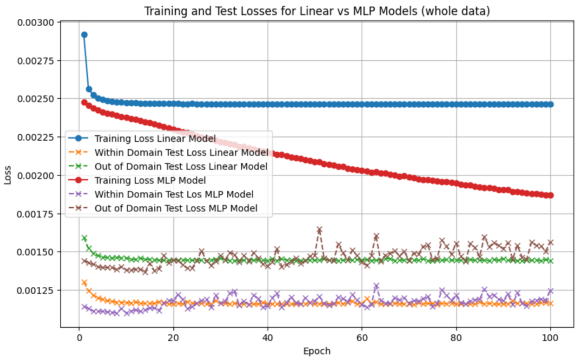

Figure 3: Epoch vs Losses for Linear Model and MLP model on whole dataset

5.1 Implementation and Setup

The implementation of our project from data fetching and data processing, to model training were all implemented by ourselves. The data preprocessing and train/test splitting were all implemented using PySpark, Numpy, and Pandas. Model architecture was done using PyTorch Module and data loading and training was done with PyTorch's DataLoader. All models were trained using a batch size of 64, learning rate of 1e-5, and Adam. We also used 100 epochs to train each model so we could capture and save the losses of each model on the different test sets.

5.2 Results

The losses for the linear and MLP regression head models trained on the full dataset are shown in Figure 3. It is clear that the linear model stops progressing after 10 epochs in both training loss, within domain loss, and out of domain loss. The MLP model on the other hand keeps improving on training loss throughout the 100 epochs. However MLP model's test losses fluctuate, showing that the model is overfitting on the training data and not generalizing on the test sets.

Figure 4 shows the training and testing results of the linear and MLP regression heads on the refined train and test datasets. Overall, the losses are lower on this refined dataset than on the full dataset, and the MLP regression head seems to overfit much more. The within domain and out of domain losses however seem to follow similar trends on both the linear and MLP regression head when compared to the full dataset.

From figure 5 we can see that the LSTM has very high out of domain test loss when compared to training and within domain test loss. The rela-

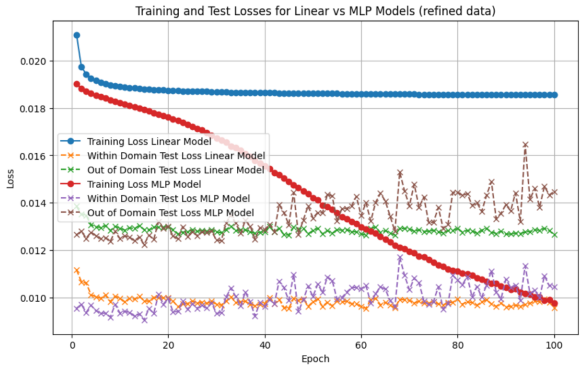

Figure 4: Epoch vs Losses for Linear Model and MLP model on refined dataset

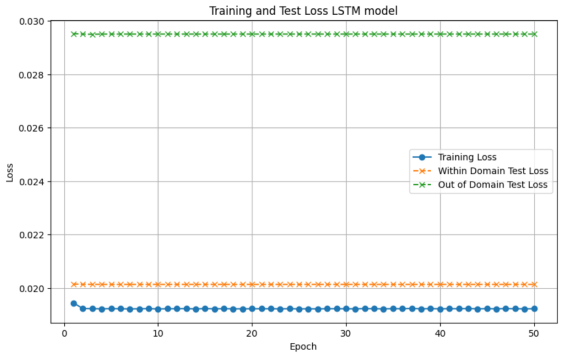

Figure 5: Epoch vs Losses for LSTM Model on refined dataset

tively flat losses indicate that the model is able to learn within the first few epochs, but does not learn afterwards.

From the first chart from figure 6 we can see that the linear regression head's predictive values are linear, which implies that it guesses the same number for all inputs. This MLP regression head had a similar trend which is why we decided to refine our dataset to try and prevent this issue.

Figures 6 shows the linear and MLP models being trained on the full and refined datasets and being tested on the refined test sets. From these figures we can see that the linear model trained on the full dataset predicts close to zero values for almost all inputs, much like our original issue. Once we train it on the refined dataset, it has more sparsity, but does not change its prediction patterns indicating that it is still poorly fit. This trend also occurs with the MLP model. When trained on the full dataset, it has some higher predicted values, but still tends towards zero. Once we train it on the refined dataset however, it makes much more accurate predictions for comments with a higher ground truth value. An Issue with this model is that

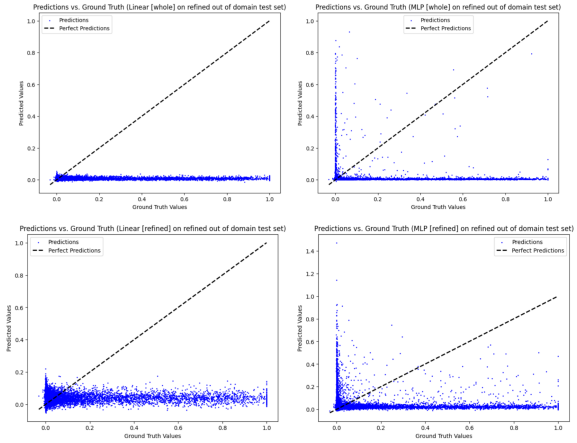

Figure 6: Prediction vs Truth of our two model trained on different dataset

MLP (whole)	MSE	RMSE	MAE	R^2
Within Domain	0.02290	0.15133	0.04738	-0.14283
Out of Domain	0.03327	0.18240	0.06361	-0.14280

Table 2: Test errors for MLP model trained on full dataset and tested on refined test data

it often predicts low score ratio comments as larger than zero, which causes this model to have a high overall error.

Tables 2 and 3 shows the errors of the MLP model when trained on the full dataset and the refined dataset, and tested on the refined test data. Despite our efforts to help the model learn to predict non zero score ratios, the errors show that this may negatively impact the accuracy of the model. The MLP trained on the refined dataset performs slightly worse across all error metrics, and the negative R2 values across each test set indicate extremely poor model fit.

Across each variation of the models training and test set, we can see some surprising results in Ta-

MLP (refined)	MSE	RMSE	MAE	R^2
Within Domain	0.02314	0.15212	0.06386	-0.15483
Out of Domain	0.03337	0.18269	0.08115	-0.1465

Table 3: Test errors for MLP model trained on refined dataset and tested on refined test data

Model (trained on) \ MSE	Within Domain Test (Whole)	Out of Domain Test (Whole)	Within Domain Test (Refined)	Out of Domain Test (Refined)
Linear (whole)	0.0022	0.00379	0.0211	0.0316
MLP (whole)	0.0024	0.00416	0.02290	0.03327
Linear (refined)	0.0040	0.00535	0.0207	0.030
MLP (refined)	0.0046	0.0076	0.02314	0.03337
LSTM (refined)	0.0042	0.0055	0.0253	0.034

Table 4: Overall MSE of different models on different datasets

Title	What was supposed to be "The Next Big Thing" but ended up becoming a flop?
Comment	watchdogs.
Prediction	0.926434
Truth	0.001

Sample high prediction

Title	Reddit- What is something that businesses throw in extra, or give "free of charge" that you do NOT like?
Comment	Email, antivirus, and file storage with internet service. Just give me my got-dang tubes and stop shoving that shit down my throat. Oh, and if I have to install a program to set up the internet, you have failed, ISP.
Prediction	0.259809
Truth	0.243187

Sample accurate prediction

Table 5: Sample Predictions

ble 4. The best performing model on the full test dataset is the linear model trained on the full dataset and the best performing model on the refined test dataset is the linear model trained on the refined dataset. This shows that while the linear model has the lowest MSE, the predictions for it will be inaccurate unless you feed it close to zero score ratio comments. This table also shows a relatively stable MSE difference between models for within domain and out of domain test sets.

5.3 Sample Predicitons from MLP model

Some sample predictions from the MLP model are seen here in Table 5. We settled on choosing these to display because both high predictions and accurate predictions are rare and are informative about the model. A common trend in high prediction comments are comments that are usually negative and either 1-2 words or 2-4 sentences. These high predictions however usually have high error as well since the comment usually has a very low ground truth value. The most accurately predicted comments often start with a single word or short sentence, but can vary greatly in length.

6 Analysis

6.1 Model Selection and Epoch's

From the results of training, it's clear that the types of predictions we get and model accuracy is heavily dependent on both the types of models we choose and the number of epochs they are trained on. Despite the MLP's training loss being low at high epochs, overfitting harms its performance in both out of domain and within domain testing. Additionally, despite the linear model's low MSE, it is not intelligently predicting comment's score ratios, and would perform poorly on more evenly distributed data. We also learned that implementing our own model using LSTM to replace DistilBERT performs similarly to the linear regression head.

6.2 Dataset quality

The decision to refine the dataset, aimed at mitigating the models' bias towards predicting low comment scores, reveals the significant impact of dataset composition on model performance. While the refinement process introduces a more balanced score distribution, it also complicates the models' ability to generalize, as evidenced by the overall performance degradation of all models when tested on both the within and out of domain dataset. However it was surprising how much worse the within domain errors were when compared to the training loss. It was our initial hypothesis that within domain loss would be more than the training loss, but not to the degree that it occurred during our experiments.

6.3 Prediction patterns

The uniform prediction trend observed in both the linear model and the MLP model (to a lesser degree) suggests a limitation in both models' ability to capture nuanced understandings of comment popularity. While analyzing the prediction samples, we can see some basic patterns for MLP predictions, however they mostly relate to high level characteristics like structure and sentiment.

7 Conclusion

The findings from this study underscore the capability of DistilBERT combined with different regression heads in predicting the popularity of comments on social media platforms such as Reddit. The MLP regression head generally outperformed the baseline linear model, indicating the advantages of using non-linear transformations and dropout in handling the complexities of natural language data. Although the LSTM model did not significantly surpass the MLP in terms of performance, its ability to process sequences might be

beneficial in different or expanded contexts. The refined dataset, which limited the dominance of less popular comments, proved essential in training the models more towards middle-range and popular comments, therefore avoiding the bias toward predicting lower scores. We found the importance of dataset composition in training machine learning models, particularly in refining input features and exploring more well constructed model architectures.

7.1 Future Work

Future work could include training DistilBERT in addition to the regression head. This would take significantly more resources, but should also significantly improve results. However the most prevalent issue with our project was the poor quality of the training dataset. It was extremely skewed which decreased the models ability to learn, so the first solution in the future would be to choose higher quality training data. This training data could be obtained from other subreddits like "NoStupidQuestion", "AskMath", "AskPhysics", etc. which could also be used to validate the model's effectiveness across various contexts and settings. It may also be beneficial to use a BERT model pretrained on QA style datasets. This could reduce the amount of fine tuning required for accurate results on our current datasets and maybe even prevent the need for higher quality training data.

References

Anthony Galtier. 2021. Fine-tuning bert for a regression task: is a description enough to predict a property's list price? *ILB Labs publications*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. Cite arxiv:1907.11692.

Louis Martin, Benjamin Muller, Pedro Javier Ortiz Suárez, Yoann Dupont, Laurent Romary, Éric de la Clergerie, Djamé Seddah, and Benoît Sagot. 2020. Camembert: a tasty french language model. In *ACL*, pages 7203–7219. Association for Computational Linguistics.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter.

Ruosong Yang, Jiannong Cao, Zhiyuan Wen, Youzheng Wu, and Xiaodong He. 2020. Enhancing automated essay scoring performance via fine-tuning pre-trained language models with combination of regression and ranking. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1560–1569, Online. Association for Computational Linguistics.