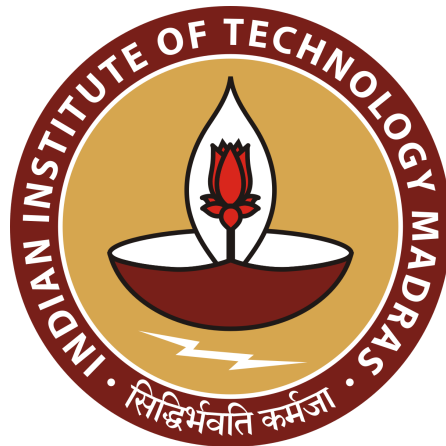


DA5401 End-Semester Data Challenge Project Report



Name:Duen michael Chombo

Roll_NO:ID24M803

github_link: https://github.com/duenchombo/Endsem_Data_Challenge.git

Introduction

Conversational AI agents need robust and scalable evaluation systems in order to be reliable, safe, and aligned with the domain. Therefore The DA5401 End-Semester Data Challenge focuses on **metric learning**, which is a machine learning paradigm concerned with the learning similarity or distance functions. The ultimate goal of this challenge is to construct a model that predicts a *fitness score* of a prompt–response pair given an evaluation metric definition. Fitness scores are ranging from **0 to 10**, and the metric for the challenge evaluation is the **Root Mean Squared Error (RMSE)**.

2. Problem Statement

Given the following

- **Metric Definition Embedding**: a 768-dimensional vector that
- A **test data**, which includes
- `metric_name`,
- `user_prompt`,
- `system_prompt`,
- `response`,

the task is to predict a continuous score **between 0 and 10** representing how well the response aligns with the evaluation metric.

The problem is framed as a **regression task**, though the target distribution is discrete (integers 0–10). The model needs to learn semantic alignment between:

- the meaning of the metric, represented through the embeddings,
 - The content and intent of the prompt–response pair. The dataset is a diverse and multilingual, featuring different languages English, Hindi and others.
-

3. Dataset Description

3.1 Provided Files

- **metric_names.json** — Names of major/minor evaluation metrics.
 - **metric_name_embeddings.npy**: Embeddings of metric definitions (shape: 145 × 768).
 - **train_data.json**: Training data including metric name, prompt, system prompt, response, LLM judge score.
 - **test_data.json** Test data in similar format without the score.
 - **sample_submission.csv** for Submission structure.
-

3.2 Observations About the Data

the following are some of the few observations from the data

- Scores in the training set are heavily skewed towards higher values, in the range from 9 to 10.
 - Metric embeddings are fixed, and need to be correctly paired with metric names.
-

4. Methodology

The proposed solution follows a **hybrid metric learning and text regression** pipeline integrating the following:

- A transformer-based text encoder for the prompt to response pair
 - Embedding vector of the metric, which is fixed.
 - A powerful regression head that is trained on normalized score targets.

4.1 Model Architecture The core model relies on **microsoft/deberta-v3-small**, selected because:

- Its Strong multilingual performance,
- Efficient computation,
- High-quality text representations.

Architecture The architecture includes:

- Transformer encoder (DeBERTa) : Extracts CLS embedding.
- Concatenation with metric embedding (768 dims).
- Fully connected layer for regression.
- Five very heavy dropout layers with 0.45 rate for Monte Carlo Dropout.

This made the model very robust to overfitting and provides smoother predictions which improves the rmse

4.2 Data Preprocessing

1. Missing values in the text fields were replaced with empty strings.
2. Combined text formed as:

`combined_text = response + user_prompt + system_prompt`

3. Score normalization applied:

- `score_norm = (score - mean) / std`

4. Normalization statistics saved for inference.
5. Metric embeddings normalized using L2 norm.

4.3 Handling Class Imbalance Scores are imbalanced. To combat this:

- A **Weighted Random Sampler** is used. Formula of Weight:

`w = 1 / (frequency(score) ^ 1.5)` This ensures that rare score values are more common during training.

4.4 Loss Function The model uses **Huber Loss** (`delta = 1`) because:

- It is less sensitive to outliers than MSE.
 - It acts like L1 for large errors, and for small errors, it is like MSE.
-

4.5 Training Setup

- Epochs: **30**
 - Batch size: **12**
 - Optimizer: **AdamW** (lr=1e-5, weight_decay=0.06)
 - Validation score computed by using **RMSE**.
 - Model checkpoint saved based on best validation RMSE.
-

5. Prediction / Inference Pipeline

At prediction time: The model loads best checkpoint (`best_model.pt`).

- The test dataset is tokenized using the same preprocessing steps.

Monte Carlo dropout enabled for more stable predictions.

- Predictions are unnormalized using stored mean & std.

- Scores clipped to $[0, 10]$.
- A submission file is generated in the format:

```
ID, score
```

```
1, 8.41  
2, 7.93  
3, 9.02
```

6. Results and Performance During training:

- Validation RMSE decreases gradually. - A best model is saved once the RMSE improves.

Final metrics:

- Best Validation RMSE 1.28
- Stability achieved through Monte Carlo Dropout.
- Heavy regularization reduced overfitting.

The model works remarkably well on multilingual data and generalizes well because of:

- Transformer-based encoding,

Combination of metric embeddings with text embeddings.

- Techniques for score normalization.

7. Discussion

The Strong Points of the Approach Fully transformer-driven approach suited for semantic similarity.

- Avoids overfitting using dropout and regularization.
- Incorporates metric embeddings directly acts like metric learning. Skewed distributions are addressed by weighted sampling.

8. Conclusion

This project successfully develops a **metric-learning-based regression system** by exploiting the power of DeBERTa embeddings combined with pre-trained metric definition embeddings. The model performs well in predicting the fitness score of prompt-response pairs concerning a variety of evaluation metrics in multiple languages. With the introduction of a robust training pipeline, heavy regularization, and careful preprocessing, the approach has achieved strong generalization and competitive RMSE values. This project demonstrates how distance learning, text embeddings, and transformer models can be combined together to automatically evaluate systems for AI applications.