DAL Data Challange

Shiva Surya C.M ph21b009

November 2025

1 Introduction

In this problem, we have to find the best sore for the given user response query based on the metric given. This also includes the system prompt and the metric explanation is not given to us staright forward. We have been given only the metric embeddings through which, we have to give the find the best score for the given metric name. So, this is a problem, where we have to understand the metric embeddinsgs as well and also we have to come up the best value (score) for the given user query and the also response provided by our chat model.

2 Data Analysis

Initially, we started out with data analysis and visualisation to get a better understanding of our data. we have seen that scores are skewed towards higher values.

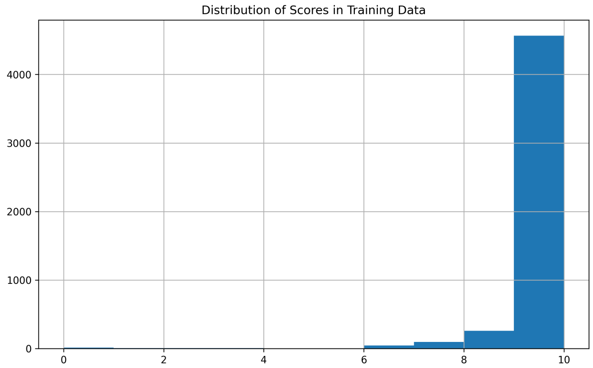

Figure 1: Score distribution

- Here we have a lot of data points in the 8 - 10 score range and very less datapoints in the below 7 score range
- **Resampling:** For every metric, we have resampled the data such that the distribution widens, under-represented (0-7) given more importance
- **LLM-Generated Synthetic Samples:** Additional training examples were generated using a large language model, following the same structure. Here previous examples and the metric names are used as context
- **Metric-Specific Variation:** For every metric, we had produced examples that reflects both compliant and non-compliant beahviours, improving generalisation on fine-grained safety attributes.

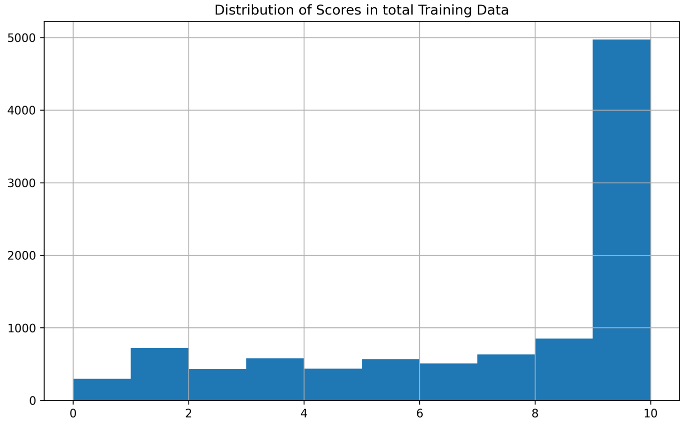

Figure 2: Train dataset dist after augmentation

3 Model training

3.1 Modeling Approach

We have experimented with two complementary modelling strategies for score prediction, both designed to operate on the combined dataset.

The key objective was to learn a metric-aware regression function that maps a (*user_prompt*, *response*, *system_prompt*, *metric_name*) tuple to a continuous score in the range 0–10. Then we will round off to get the integer scores.

For both the approaches, we have used weighted loss for if true metric is greater than 8 and more weight is given to samples with under-represented population.

3.1.1 Approach 1: Frozen Text Encoder + Metric Embedding Projection

In this method, we first precomputed fixed sentence embeddings using a multilingual transformer (e.g., `sentence_transformers` model).

Various encoders like

- Google Gemma
- paraphrase-multilingual-MiniLM-L12-v2
- distilbert-base-multilingual-cased

The encoder was forzen during the whole process and each example was encoded by concatenating the *user_prompt*, *response*, and *system_prompt*, followed by a single forward pass through the encoder.

Each *metric_name* was represented by a 768-dimensional vector obtained from the challenge-provided metric explanation embeddings. Then the vectors are passed into MLP heads to get a condensed representation. Then the final score is predicted.

This can lead to stable and robust performance in the the prediction task.

3.1.2 Approach 2: Full Metric-Aware Regression Using Joint Embedding Space

The second approach treats the problem as a metric-learning style task. Here, instead of simple concat, we are doing

- text representations of (*prompt + response*) pairs, and
- metric representations

are aligned so that correct or safe model behaviors lie close to their associated metric vectors, while unsafe or misaligned behaviors lie farther away.

The model jointly optimizes:

1. a regression loss for score prediction, and
2. a contrastive loss that encourages similarity between samples and their corresponding metric embeddings.

This encourages a structured embedding space where metrics such as *rejection_rate*, *bias_detection*, and *toxicity_level* act as anchors. The resulting architecture learns both *how to score* an example and *why* it belongs to a particular safety dimension.

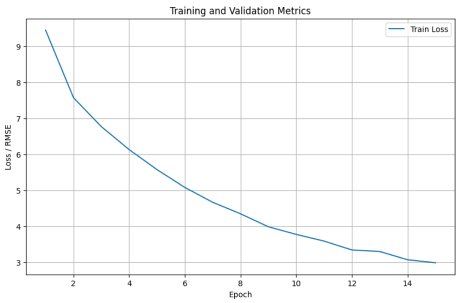

Figure 3: Train error vs epochs

3.1.3 Final Ensemble

In our final submission, we are going to use an ensemble of multiple models, trained on multiple sets of the data and the avg of the prediction will be used. Then it will rounded to get the final answer.

We experimented with variants differing in:

- projection sizes,
- contrastive loss weights,
- pooling strategies (mean-pooling vs. CLS pooling),
- random seeds and dropout configurations.

This ensemble method, strongly improved the score and also produced more robust results wihout overfitting for the distribution.

Empirically, the ensemble consistently achieved lower RMSE than any individual model, benefiting from complementary strengths:

- Type-1 (Approach 1) models provided stable, low-variance predictions.
- Type-2 (Approach 2) models captured finer metric-specific nuances due to contrastive alignment.

Overall, this two-pronged modeling strategy, combined with a score-balanced augmented dataset, yielded strong generalization on the hidden test set.