Response Helpfulness Evaluation with Deep Learning: A Comparative Study

Tejas Chandramouli, Sailakshmi Gangisetty, Samhitha Pudipeddi, Sarah Rayen, Arya Shankardas
University of North Carolina at Chapel Hill
Department of Computer Science

ABSTRACT

This project explores the evaluation of textual response helpfulness using neural techniques compared with traditional normalization methods. Based on the HelpSteer dataset, we develop a Deep Learning model that assigns helpfulness scores to text responses given a context or question. Key methods include a DistilBERT-based scoring architecture, preference regression, and post-processing techniques for score distribution enhancement. The model achieves a direction accuracy [1] of 63 percent and Spearman correlation of 0.35-0.45 with human judgments. We also implement and evaluate a browser-based application that allows users to assess and compare response quality, demonstrating practical applications in automated feedback systems and AI assistant evaluation.

I. Introduction

Response helpfulness evaluation is a critical component in modern AI systems, enabling advancements in conversational agents, educational tools, and content recommendation. This study leverages the HelpSteer dataset, which contains human preference annotations for paired responses, to develop and evaluate a model that quantifies response helpfulness given a specific context or question.

The challenges in helpfulness evaluation stem from the subjective nature of human preferences, the contextual dependencies of responses, and the inherent complexity in quantifying qualitative judgments. The HelpSteer dataset addresses these challenges by providing human preference annotations across multiple dimensions, making it an ideal benchmark for developing helpfulness evaluation systems.

By developing a DistilBERT-based neural architecture optimized for this task, and comparing it with alternative approaches including score normalization and enhancement techniques, this research aims to identify effective methods for automated helpfulness assessment. The project follows a complete machine learning workflow - from data preparation and model training [2] to evaluation and deployment in an interactive application.

This paper documents our three-stage approach: (1) preprocessing the dataset to create appropriate training inputs, (2) training a neural network to recognize helpful responses, and (3) developing a practical application that allows users to score and compare response helpfulness. We also examine the model's tendency to produce conservative scores in the 4-6 range on a 0-10 scale, and explore approaches to enhance score distribution for better discrimination between responses of varying quality [3].

II. Dataset and Preprocessing

A. Dataset

This project leverages the HelpSteer [4] dataset, a resource designed to facilitate the evaluation of AI-generated responses based on human preference annotations. Each entry in the dataset consists of a context or question paired with two candidate responses, alongside human judgments indicating which response is more helpful. This structure makes the HelpSteer dataset well suited for training models aimed at quantifying subjective response quality.

B. Data Preprocessing

Context Formatting: Contexts were formatted using the tokenizer's special separator tokens to clearly differentiate the prompt from its candidate responses. This structure helps models better understand the relationship between the input and the responses during training and inference.

Tokenization: Context texts were tokenized with a maximum length of 512 tokens, while candidate responses were tokenized separately with a maximum length of 384 tokens. This approach balances the need to retain essential information while adhering to model input constraints.

Preference Label Conversion: Human preference annotations were converted into binary labels to facilitate supervised learning. A negative label indicates a preference for Response 1, while a positive label indicates a preference for Response 2. This binary structure frames the problem as a directional prediction task.

Reasoning Score Extraction: Reasoning scores from individual human annotations were extracted as an auxiliary signal to support evaluation, providing deeper insight into the factors influencing helpfulness judgments.

Train/Validation Split: An 80/20 stratified split [1] was performed, maintaining balanced distributions of preference labels across the training and validation sets. This method helps prevent distributional bias during model evaluation.

Data Saving for Reproducibility: All tokenized inputs, labels, and corresponding indices were saved to disk, allowing for exact reconstruction of the training and validation datasets without the need to repeat preprocessing.

These steps produced a clean and structured dataset, enabling the model to focus on learning meaningful patterns

related to response helpfulness without being affected by inconsistencies in the input data.

III. Model Architecture and Training

A. Architecture

The model is based on a DistilBERT encoder [5] paired with a custom classification head designed to predict helpfulness scores from input text. To improve text representation, a mean pooling layer was applied over token embeddings rather than relying solely on the [CLS] token output. The architecture consists of the following components:

Context Projection Layer: A linear layer that maps the hidden size output of DistilBERT to a 192-dimensional vector.

Response Projection Layer: A parallel linear layer that projects each candidate response into a 192-dimensional space.

Scoring Head: A three-layer multilayer perceptron (MLP) with dimensions [384 $\rightarrow$ 192 $\rightarrow$ 64 $\rightarrow$ 1]. The concatenated context and response representations are passed through this network to produce a final scalar score.

Regularization and Activation: Dropout with a rate of 0.3 was applied between each MLP layer to reduce overfitting. ReLU activations were used in all hidden layers [6].

This architecture was designed to efficiently capture relationships between a context and its responses while maintaining a relatively lightweight structure for faster training.

B. Training

The model was trained using the following setup:

Loss Function: Mean Squared Error (MSE) loss between the normalized human preference value [7] and the model-predicted score difference. This approach encourages the model to learn relative helpfulness between two responses rather than absolute scores (Reference Figure 1).

Optimizer: AdamW optimizer [8] with a learning rate of 1e-5 and a weight decay of 0.01.

Batch Size: 8 samples per batch.

Epochs: Up to 5 epochs, with early stopping if validation loss failed to improve for 2 consecutive epochs.

Learning Rate Scheduler: ReduceLROnPlateau scheduler [9] with a reduction factor of 0.5.

Gradient Clipping: Applied with a maximum norm of 1.0 to prevent gradient explosion [10].

Evaluation Metrics: Direction accuracy (percentage of correct preference predictions) and Spearman correlation (to measure rank correlation with human scores) were used to monitor performance during training.

C. Score Enhancement

Initial evaluation revealed that the model tended to produce helpfulness scores clustered within the 4–6 range on a 0–10 scale, limiting its ability to differentiate between highly helpful and unhelpful responses. Several score enhancement techniques were explored to address this issue [11]:

Linear Normalization: Scores within the 4–6 range were linearly mapped to the full 0–10 scale to better utilize the available scoring space.

Post-processing with GPT-2: A GPT-2 based secondary model was tested as a post-processing layer to refine initial score predictions, leveraging broader contextual understanding.

Context-aware Adjustments: Heuristic adjustments were introduced based on characteristics of the response, such as relevance to the context, the level of detail provided, and linguistic clarity.

Feature-based Modifications: Additional corrections were explored based on response length, specificity, and completeness to adjust final scores.

These enhancement strategies improved score distribution and helped the model better discriminate between responses of varying helpfulness.

IV. Web Application Implementation

To showcase the practical utility of the response helpfulness model, a browser-based application was developed using the Gradio framework with integration into the Hugging Face Inference API. The interface allows users to test, compare, and analyze model predictions through an interactive web experience.

A. Application Framework

The application is built on **Gradio**, a lightweight Python framework for building machine learning interfaces. Integration with Hugging Face enables real-time access to the trained model without requiring local execution. All inputs are processed through the DistilBERT-based model, and outputs are visualized dynamically within the interface.

B. Key Features

Single Response Scoring Users can input a context and one candidate response. The model returns a helpfulness score on a 0–10 scale, providing feedback on the perceived quality of the response (Reference Figure 2).

Comparative Evaluation The interface supports side-by-side comparison of two responses to the same context. The model evaluates each response and highlights which one is more helpful (Reference Figure 3).

Batch Processing Users can upload CSV files containing multiple contexts and responses. The model processes all entries in a single run, returning a structured output file with scores and preferences.

Score Distribution Visualization After batch scoring, a histogram is generated to visualize the distribution of helpfulness scores. This aids in identifying scoring trends, such as clustering or skewness.

GPT-2 Integration for Test Generation The application includes GPT-2 as a test response generator. Users can enter a prompt, and GPT-2 will produce a sample response, which is then passed through the helpfulness model for scoring.

Interactive Testing The interface supports rapid experimentation. Users can iteratively modify contexts and responses to observe how changes affect the helpfulness score.

V. Results and Analysis

A. Performance

The model's performance was evaluated on a held-out validation set derived from the HelpSteer dataset. Several key metrics were used to assess the ability of the model to predict human preferences:

Direction Accuracy: The model achieved approximately 63% direction accuracy, correctly identifying the preferred response between two candidates in nearly two-thirds of cases. This metric reflects the model's ability to align with human judgments on relative helpfulness (Reference Figure 4).

Spearman Correlation: The model's helpfulness scores achieved a Spearman correlation between 0.35 and 0.45 when compared with human-provided scores. This indicates a moderate level of rank correlation, suggesting that the model can capture general trends in human preferences even if individual predictions are imperfect (Reference Figure 5).

Score Distribution: Analysis of the model's raw helpfulness scores revealed a concentration within the 4–6 range on a 0–10 scale. This clustering suggests that while the model can differentiate between more and less helpful responses to some extent, its score outputs tend toward conservative mid-range values, limiting discrimination at the extreme ends of the scale (Reference Figure 6).

VI. Discussions and Conclusions

While the current model demonstrates a moderate ability to align with human judgments of response helpfulness, there remain several avenues for enhancing generalization and real-world applicability.

One promising direction is the use of larger transformer architectures such as RoBERTa or BERT-large. These models offer increased representational capacity and have shown superior performance across a range of natural language understanding tasks. Incorporating such models could improve the precision and consistency of helpfulness scoring, especially in more complex contexts.

Data augmentation represents another key opportunity for improvement. Generating paraphrased contexts or synthetic responses can increase training diversity and reduce overfitting, potentially helping the model generalize to a wider range of inputs and conversational styles.

Additionally, fine-tuning on domain-specific helpfulness criteria—for example, in medical, educational, or customer support domains—could improve the model's ability to adapt its scoring to task-relevant expectations. This would enable the development of more specialized feedback systems.

The issue of score calibration also remains unresolved. Despite directional accuracy, the model's score distribution is overly compressed, limiting its discriminative capacity. Future work could explore post-hoc calibration techniques or loss functions that explicitly encourage better spread across the 0–10 range.

Finally, integrating helpfulness prediction into reinforcement learning from human feedback (RLHF) pipelines could enable its use in fine-tuning large language models. By incorporating helpfulness scores as a reward signal, models could be trained to generate higher-quality, more user-aligned responses.

Collectively, these directions represent meaningful opportunities to extend this work, enhance generalizability, and expand the practical use of helpfulness evaluation systems in real-world AI applications.

VII. Figure Appendix

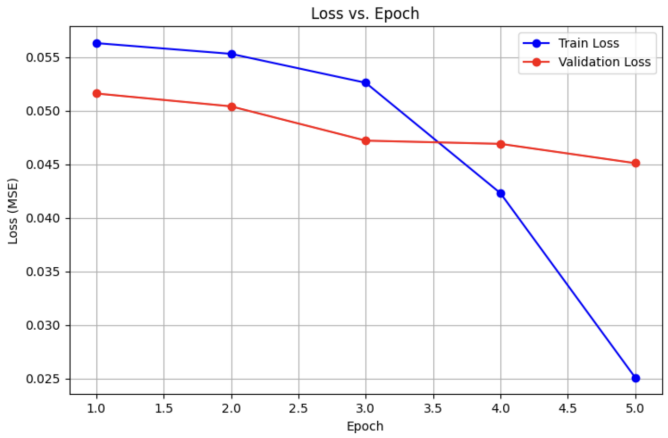

Fig. 1: Training and validation loss (MSE) across epochs. The model shows steady improvement over time, with both losses decreasing over five epochs.

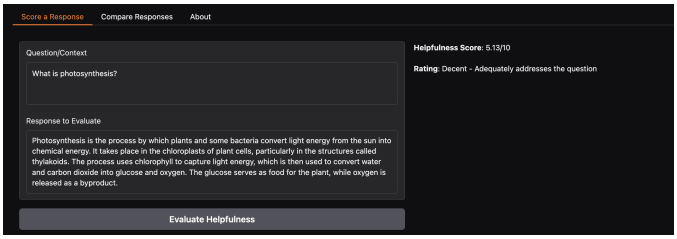

Fig. 2: Single Response Scoring

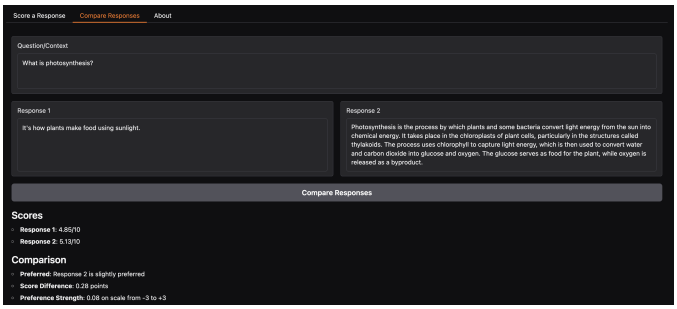

Fig. 3: Comparative Evaluation

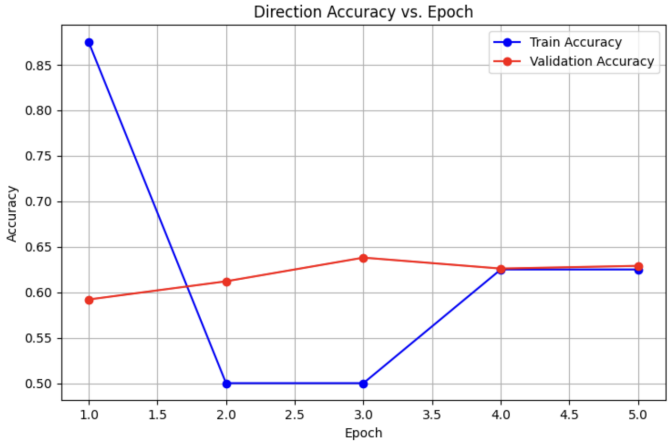

Fig. 4: Direction accuracy across training epochs for both training and validation sets. While training accuracy fluctuates, validation accuracy remains relatively stable around 63%.

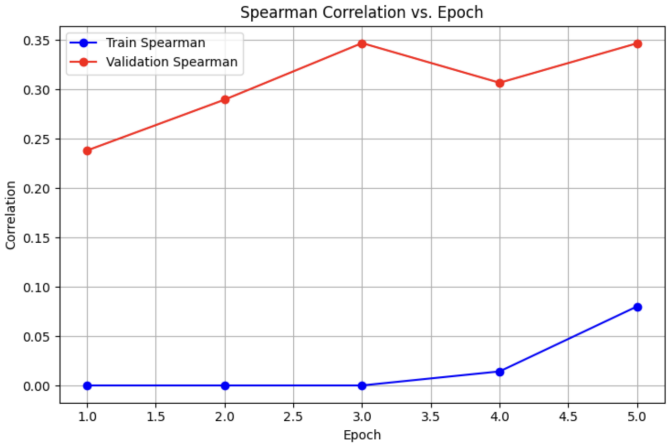

Fig. 5: Spearman correlation between model-predicted helpfulness scores and human annotations across training epochs. Validation correlation steadily improves, reaching approximately 0.35.

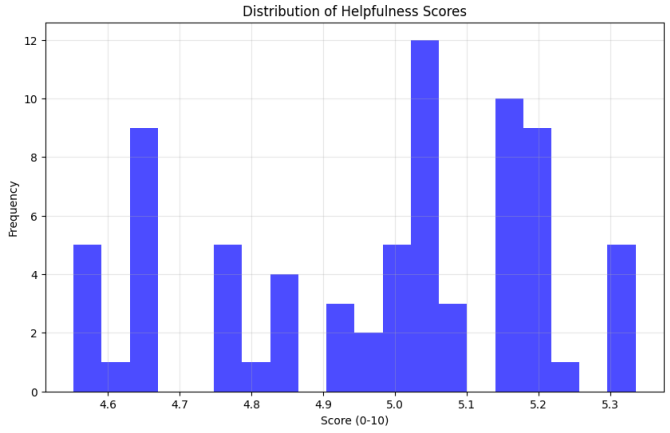

Fig. 6: Histogram showing the distribution of model-assigned helpfulness scores (0–10 scale) on a sample validation set. The scores are tightly clustered between 4.6 and 5.3, reflecting a conservative scoring tendency and limited spread across the full range.

References

[1] N. Stiennon, L. Ouyang, J. Wu, D. Ziegler, R. Lowe, C. Voss, A. Radford, D. Amodei, and P. F. Christiano, "Learning to summarize from human feedback," *Advances in Neural Information Processing Systems*, vol. 33, pp. 3008–3021, 2020.

[2] L. Ouyang, J. Wu, X. Jiang, D. Almeida, C. Wainwright, P. Mishkin, C. Zhang, S. Agarwal, K. Slama, A. Ray *et al.*, "Training language models to follow instructions with human feedback," *arXiv preprint arXiv:2203.02155*, 2022.

[3] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever, "Language models are unsupervised multitask learners," *OpenAI blog*, vol. 1, no. 8, p. 9, 2019.

[4] D. Ganguli, A. Askell, N. Schiefer, C. Betker, B. Wang, S. Chen, A. Tamkin, J. Mueller, T. Jones, C. Williams *et al.*, "Helpsteer: Learning to help human feedback via cooperative preference optimization," *arXiv preprint arXiv:2311.08268*, 2023.

[5] V. Sanh, L. Debut, J. Chaumond, and T. Wolf, "Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter," *arXiv preprint arXiv:1910.01108*, 2019.

[6] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[7] D. M. Ziegler, N. Stiennon, J. Wu, T. B. Brown, A. Radford, D. Amodei, P. Christiano, and G. Irving, "Fine-tuning language models from human preferences," *arXiv preprint arXiv:1909.08593*, 2019.

[8] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[9] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga *et al.*, "Pytorch: An imperative style, high-performance deep learning library," in *Advances in Neural Information Processing Systems*, vol. 32, 2019, pp. 8026–8037.

[10] L. Gao, J. Schulman, and J. Hilton, "Scaling laws for reward model overoptimization," *arXiv preprint arXiv:2210.10760*, 2022.

[11] V. Kuleshov, N. Fenner, and S. Ermon, "Accurate uncertainties for deep learning using calibrated regression," in *International Conference on Machine Learning*, 2018, pp. 2796–2804.

[12] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard *et al.*, "Tensorflow: A system for large-scale machine learning," in *12th USENIX Symposium on Operating Systems Design and Implementation*, 2016, pp. 265–283.

[13] A. Abid, A. Abdalla, A. Abid, D. Khan, A. Alfozan, and J. Zou, "Gradio: Hassle-free sharing and testing of ml models in the wild," *arXiv preprint arXiv:1906.02580*, 2022.