Response Helpfulness Evaluation with Deep Learning: A Comparative Study

Tejas Chandramouli
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 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 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.

II. Dataset and Preprocessing

Talk about this

Data Source: HelpSteer dataset (pairs of responses with human preference annotations)

Key Preprocessing Steps:

Format context text with proper separator tokens using tokenizer's special tokens

Tokenize contexts with 512 max tokens; responses with 384 max tokens Convert preference values to binary/relative values (negative = response 1 preferred, positive = response 2)

Extract reasoning scores from individual annotations

Create stratified 80/20 train/validation split to ensure balanced preference distribution

Save tokenized data and indices for reproducibility

A. Data Preprocessing

III. Model Architecture and Training

A. Architecture 1

Talk about this

Base Model: DistilBERT encoder with custom classification head Key Components:

Mean pooling layer (instead of just CLS token) for better text representation Context projection layer: Linear(hiddensize $\rightarrow$ 192) Response projection layer: Linear(hiddensize $\rightarrow$ 192) Scoring head: 3-layer MLP with dimensions [384 $\rightarrow$ 192 $\rightarrow$ 64 $\rightarrow$ 1] Dropout (0.3) between layers for regularization ReLU activations in hidden layers

1) Training 1: Training

Loss Function: Preference Score Difference MSE Loss (normalized preference values vs. score differences) Optimizer: AdamW with learning rate 1e-5 and weight decay 0.01 Training Strategy:

Batch size: 8 samples Training epochs: 5 with early stopping (patience=2) Learning rate scheduler: ReduceLROnPlateau with factor=0.5 Gradient clipping (maxnorm=1.0) Direction accuracy and Spearman correlation as evaluation metrics

2) Performance and Analysis: Score Enhancement Techniques

Original Score Range Issue: Scores clustered in 4-6 range on 0-10 scale Enhancement Approaches:

Linear normalization: Map [4-6] range to [0-10] GPT-2 based evaluation as post-processing layer Context-aware

score adjustment using response characteristics Feature-based adjustments (length, detail level, relevance)

IV. Results and Analysis

Web Application Implementation

App Framework: Gradio interface with Hugging Face integration Key Features:

Single response scoring Comparative evaluation between two responses Batch processing capability Score distribution visualization Integration with GPT-2 for generating test responses Interactive testing interface

A. Performance

Model Performance

EMNIST Test Set Metrics:

Direction accuracy: 63Spearman correlation: 0.35-0.45 with human judgments Score distribution: concentrated in 4-6 range out of 10

B. Generalizations

V. Discussions and Conclusions

Visualizations to Include

Key Figures:

Score distribution histogram Loss curves during training Comparison of original vs. adjusted score distributions Sample UI screenshots from the Gradio application Correlation plots between model scores and human preferences

Future Work

Enhancement Points:

Explore larger transformer models (RoBERTa, BERT-large) Data augmentation techniques to improve generalization Fine-tuning on domain-specific helpfulness criteria Better calibration methods for score distribution Integration with RLHF pipelines for LLM training

References