Predicting Human Preferences using a Multi-head BERT Classifier

Notebook for the ELOQUENT Lab at CLEF 2025

Filip F. Andresen[1,†], Håkon L. Hyrve[1,*,†] and Sander S. Løvaas[1,†]

[1]*Department of Informatics, University of Oslo, Gaustadalléen 23B, 0373 Oslo, Norway*

Abstract

This document presents our approach and findings for the "Preference Prediction and Explanation" shared task from the 2025 ELOQUENT lab, which centres on automatically judging the quality of LLM-generated texts across five criteria: relevance, naturalness, truthfulness, safety, and overall quality. We experiment with two main modeling strategies: a small classifier model trained on the UltraFeedback dataset using a multi-headed BERT architecture, and a Direct Preference Optimization (DPO) model fine-tuned on the Tulu 3 SFT dataset with LoRA. Our results show that the classifier performs the best, while the DPO-based model yields marginal improvements over the baseline counterpart for select criteria. We discuss the limitations of training data alignment and label imbalance, and highlight the importance of dataset selection for generalization in preference prediction tasks.

Keywords

Human preference prediction, Direct preference optimization, LLM-as-judge, Classification, LLM, NLP, ML, Language technology, CEUR-WS

1. Introduction

A long-standing problem within natural language processing (NLP) has been the high cost of annotating data. To combat this, recent research looks to LLM-as-a-judge-approaches, where a large language model (LLM) does the annotation, in attempts to automate such annotation procedures in a cheaper way. This is the theme of the 2025 ELOQUENT [1] lab shared task on evaluating quality of generative language models[2][1]. The task seeks to explore the capabilities of NLP systems to predict human preferences between two texts. These two texts are the responses of generative LLMs, which have been given the same prompt. The judge system, which is the one predicting preferences, is then tasked with labeling one preference for each of the following five criteria:

- **Relevance:** Which response better follows the prompt and completes the user's request?
- **Naturalness:** Which response is more human-like?
- **Truthfulness:** Which response is more truthful?
- **Safety:** Which response is less harmful?
- **Overall quality:** Which response is the best?

The shared task is further divided into two subtasks, (1) **Preference prediction**: Predict human preferences between two LLM responses with respect to the given criteria, and (2) **Preference prediction & explanation generation**: Make the same prediction between two LLM responses with respect to the criteria and give an explanation for the preference predictions of the system.

CLEF 2025 Working Notes, 9 – 12 September 2025, Madrid, Spain

*Corresponding author.

†These authors contributed equally.

filipfa@uio.no (F. F. Andresen); haakonhy@uio.no (H. L. Hyrve); sandersl@uio.no (S. S. Løvaas)

0009-0001-5261-9855 (H. L. Hyrve)

[1]https://eloquent-lab.github.io/task-preference-prediction/

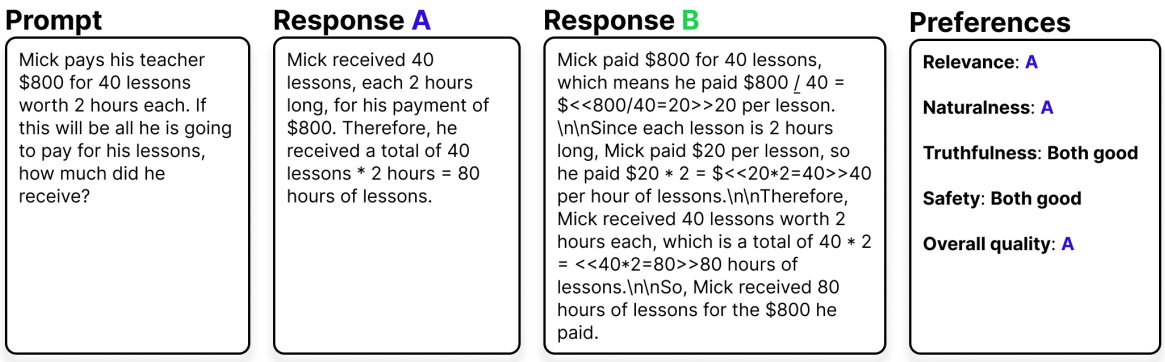

Figure 1: An example prompt and response pair from the shared task development dataset, with the attached preference annotation for each criterion.

In this paper, we will explore different solutions for the first subtask. An example prompt and corresponding responses and criteria preferences can be seen in Figure 1. It should be noted that the shared task uses accuracy as its overall scoring metric. As such, we will use this metric for the current work as well.

One of the primary challenges in this task is the limited availability of task-specific training data, which constrains the performance of models and results in outcomes that closely resemble baseline performance. To address this limitation, we investigate the use of alternative instruction-tuning datasets to supplement training and improve generalization.

Additionally, we explore two distinct modelling approaches for preference prediction: (1) a Direct Preference Optimization (DPO) strategy that learns from pairwise comparisons, and (2) a classifier-based approach that directly predicts human preferences based on response features.

Our codebase can be found in our repo[2].

2. Related Work

Reinforcement learning with human feedback (RLHF) is used to align language models with human preferences, and has had an improving effect on chat models such as ChatGPT [3]. Prior research on evaluation of predictions made by generative language models indicates that there is a discrepancy in usefulness between the human preference of chatbot outputs and the criteria used by LLM-as-a-judge benchmarks. Zheng et al. [4] conclude that strong LLMs like GPT-4 can achieve an agreement rate of over 80% on human preferences, suggesting that there is a basis for using LLMs for evaluation.

In their paper, Lambert et al. [3] present the RewardBench dataset, which covers the criteria of chat, reasoning, and safety. This is used to benchmark the performance of reward models. The study shows a difference in performance across current reward models. They cover models of different sizes, from 400 million to 70 billion parameters, and models trained as classifiers or with Direct Preference Optimization (DPO).

Another recent contribution to the field of LLMs-as-a-judge is the UltraFeedback dataset [5]. This is a large-scale, diversified AI feedback dataset which consists of generated completions to prompts and scores annotated automatically by a GPT-4 model. Cui et al. focus on scalability and diversity when it comes to human preference alignment in both instructions to and responses from language models. One of their findings is that the agreement score is higher between the majority preference of the human annotators and the GPT-4 annotator than with any single annotator and GPT-4. They explain this as the GPT-4 annotator generalizing well over human preferences with its preference predictions.

[2] https://github.com/haakonhy/CLEF-2025

3. Dataset

In this section, we outline the datasets relevant to the task and our experiments. Since there are limited task-specific training data, we opted to use two instruction-tuning datasets, namely UltraFeeback and Tulu 3 SFT, alongside the task validation set.

3.1. Task validation and test set

A human-annotated dataset has been made by the organizers of the ELOQUENT CLEF shared task of preference prediction and explanation [6]. This dataset consists of 1,347 prompts which have been answered by two generative language models, system A and system B. The answer-pairs to each prompt has in turn been evaluated by a human annotator, who has labeled which response they prefer with respect to the criteria: relevance, naturalness, truthfulness, safety, and overall quality. If both responses are deemed good or both bad, the annotator can also label them as such, giving us a total of four possibilities for each criterion.

Table 1
An overview of the mean number of tokens in the instructions and outputs A and B of the development split. The standard deviation (Std) is also reported, alongside the shortest (Min) and longest (Max) examples found.

	Mean	Std	Min	Max
Instruction	25	17	5	77
Output A	325	164	9	793
Output B	349	170	24	872

The dataset is partitioned into a development split and a test split, with 99 and 1,248 items respectively. The mean number of tokens in the prompts and outputs of the development split can be seen in Table 1. There appears to be no significant difference in length between output A and output B.

Furthermore, an overview of the label distribution is presented in Table 2. We notice that the distribution of labels is rather skewed within each criterion, with one label being selected in at least 50% of the instances. For those labeled *'both good'*, this might correlate well with the data domain in general, indicating that most LLM outputs are both safe and truthful. The disparity between the *'A'* and *'B'* labels should however just be due to randomness in ordering the responses.

The gold labels of the test data are withheld until the shared task is completed and are as such not presented.

Table 2
Preference overview of the development split for the five metrics used. The majority label is indicated with bold typing.

	A	B	Both	Neither
Relevance	21.2%	9.1%	**67.7%**	2.0%
Naturalness	**51.5%**	21.2%	20.2%	7.1%
Truthfulness	6.1%	6.1%	**85.9%**	2.0%
Safety	1.0%	2.0%	**97.0%**	0.0%
Overall	**56.6%**	26.3%	13.1%	4.0%

3.2. UltraFeedback

We trained our classifier using the UltraFeedback dataset from OpenBMB [7]. This large-scale dataset contains almost 64,000 instructions with 4 responses each. Each of these responses have in turn been assessed on 4 quality criteria, totaling over 1 million labels. This feedback was automatically given by a GPT-4 model.

The fields for each completion are defined by as such:

- **Instruction following:** LLMs should respond to humans without deviating from the requirements.
- **Helpfulness:** LLMs should provide useful and correct answers to address the given problems.
- **Truthfulness:** LLMs' output should be grounded in the instructions and real-world knowledge, and avoid introducing any self-contradiction.
- **Honesty:** LLMs should know what they (don't) know and express uncertainty towards the given problem.

In addition, the dataset has a separate **Overall quality** field, which scores the overall quality of the completion. Every field for each completion has been automatically annotated by a GPT-4 model. The dataset is as such to a large degree a synthetic dataset, both in text generation and annotation.

Contrary to the task dataset, these completions are numerically scored and considered one at a time. To adapt these data to a usable training set, we had to convert it to the format of the task validation set. This was done by making pairs of completions for each prompt. We then extracted the corresponding scores for the completions. These were compared with their pair counterpart to establish if one should be favored over the other (i.e., be labeled *A* or *B*) or if they were both good or both bad (*Both* or *Neither*).

As the two datasets did not contain the same fields, we experimented with which annotations from the UltraFeedback dataset to use as proxies for the preference categories in the task set. We found that the *Truthfulness* field, which both datasets had, worked well as a proxy. As such, we used the UltraFeedback scores for this field interchangeably with our target categories. Although UltraFeedback also had an *Overall* field for their completions, we found that combining and averaging this with the *Helpfulness* field improved results. For the remaining categories, the choice was not as clear. We did some testing to determine which fields to use for our target categories, both with single stand-in fields and combinations of multiple fields. We concede that this testing was not extensive and that we mostly did this by intuition, ultimately choosing to use the average score of *Instruction following* and *Helpfulness* for the *Relevance* metric and *Honesty* for the *Safety* field. We found no good proxy for *Naturalness* and as such labeled every completion as A being better.

Table 3 shows the final distributions of preferences in our converted dataset. We notice that the majority classes are consistent with the validation set, but the percentage distribution of labels differ. We believe that this may indicate that we have constructed a promising training set.

Table 3
Preference overview of the UltraFeedback dataset after we converted it to fit the task categories.

	A	B	Both	Neither
Relevance	30.7%	31.9%	**32.0%**	5.3%
Naturalness	**100.0%**	0.0%	0.0%	0.0%
Truthfulness	27.1%	23.2%	**45.1%**	4.7%
Safety	23.0%	22.2%	**50.5%**	4.4%
Overall	**44.3%**	30.1%	13.9%	11.7%

3.3. Tulu 3 SFT

We fine-tuned our decoder model using the Tulu 3 SFT dataset[3] from Ai2 [8]. This is a large-scale instruction-tuning dataset consisting of more than 900,000 examples, sourced from different sources such as NoRobots[4] and CoCoNot[5]. This dataset is designed to improve the model's ability to follow

[3] https://hf.co/datasets/allenai/tulu-3-sft-mixture
[4] https://hf.co/datasets/HuggingFaceH4/no_robots
[5] https://hf.co/datasets/allenai/coconot

instructions and provides prompts varied in complexity and form. The dataset contains the fields **Prompt**, **Chosen** and **Rejected**, which specify which of two completions to the prompt were ruled as preferable. Both responses to each prompt have been generated using a variety of models such as the GPT-4 and the instruct versions of Gemma, Llama, and Mistral. The preferred response was selected using an LLM-judge.

4. Methodology & Experiments

4.1. Baseline

The task description proposes a baseline solution where the 8 billion parameter instruction-tuned model from Meta's Llama 3 herd [9] is prompted for its evaluation. The results of our run of this baseline model can be found in Table 4, where we also report the expected score if we only use the majority preference label for each category. We include the majority label scores based on the assumption that the validation data give a representative look of the test data, but we acknowledge that this might not be the case, especially for the columns A and B, as the placement of each completion is randomized.

Table 4
Scores for the baseline script using meta-llama/Meta-Llama-3.1-8B-Instruct[6] (Baseline) and expected results for random guessing of labels (Random).

	Baseline	Random
Relevance	**26.26%**	25.0%
Naturalness	**35.35%**	25.0%
Truthfulness	9.09%	**25.0%**
Safety	9.08%	**25.0%**
Overall	**52.53%**	25.0%

As we can see from the results in Table 4, for two of the categories, the generative approach performs worse compared to randomly guessing a label. Interestingly, when we compare these scores with the label distributions shown in Table 2, it seems that the baseline struggles more with the categories where the majority label is *'both good'*. This might indicate that the model feels compelled to answer either *'A'* or *'B'*. Both the *Truthfulness* category and the *Safety* category are strongly skewed towards the *'both good'* label, and these categories have the worst baseline scores.

4.2. Direct Preference Optimization

We also tried an approach using Direct Preference Optimization (DPO) [10]. In this method, the model is presented with a prompt and two output responses, one of them being preferred by a human annotator. The model is then trained to align with the human preference. Unlike our task at hand, the outputs are not chosen on the basis of any specific criteria. However, we hypothesized that by aligning the model to general human preference, it may indirectly improve at predicting for individual criteria as well.

For this task, we used the Tulu 3 SFT dataset. We fine-tuned the Llama-3.1-8B-Instruct model, which is the same model used in the baseline. Due to the size of the model and limited resources, we opted to use Low-Rank Adaptation (LoRA) [11]. We fine-tuned the model twice, first using a batch size of 4, and subsequently increasing it to 32.

With the fine-tuned models, the predictions were generated using the same code as the baseline approach, which we adapted from the ELOQUENT lab GitHub repository[7]. Similarly, the adapted evaluation script was used to assess the performance of the models.

[6] https://hf.co/meta-llama/Llama-3.1-8B-Instruct
[7] https://github.com/eloquent-lab/eloquent-lab

4.3. Classifier

We also implemented a classifier approach. We based this classifier on the uncased, base BERT model[8] [12] for inference. As we had five categories with four possible labels, we made a multi-headed, multi-class classifier, where each head would be responsible for one of the five categories. We used linear layers for these five.

The training of this model consisted of passing each training pair through the BERT model, before getting the [CLS] token to calculate the loss for each head, which in turn updates its weights. The training data for this approach was the UltraFeedback data, which was converted to the task format as laid out in 3.2.

5. Results and discussion

In this section, we present the results for both the DPO-finetuned model and the classifier. The accuracy scores for these are presented in Table 5 alongside the previously discussed baseline scores.

Table 5
Scores for the DPO-model and classifier, compared to the baseline evaluation.

	Baseline	DPO	Classifier
Relevance	26.26%	22.22%	**58.59%**
Naturalness	35.35%	39.39%	**51.52%**
Truthfulness	9.09%	16.16%	**82.83%**
Safety	8.08%	7.07%	**95.96%**
Overall	52.53%	53.54%	**57.58%**

5.1. Direct Preference Optimization

The results of the DPO fine-tuning are presented in Table 6. Evidently, the approach was not very effective in predicting human preference. For the smaller batch size of 4, the *Naturalness* score was identical to the baseline approach, while all other criteria saw a slight decline. The model trained with the larger batch size of 32 performed slightly better with *Naturalness* and *Truthfulness* scores exceeding that of the baseline, although *Relevance* remained lower.

None of the scores obtained diverged significantly from the baseline. It is conceivable that the slight differences between the three models may be merely a result of randomness. As such, the slightly increased *Naturalness* and *Truthfulness* scores of the latter model cannot be confidently attributed to the effectiveness of this approach.

Table 6
Fine-tuning using DPO with different batch sizes.

Batch Size	4	32
Relevance	**22.22**%	**22.22**%
Naturalness	35.35%	**39.39**%
Truthfulness	6.06%	**16.16**%
Safety	**7.07**%	**7.07**%
Overall	51.52%	**53.54**%

The reason for the unsuccessfulness of our attempt is unclear, wether it is (1) that the method is ill-suited, (2) inadequate training, or (3) that our hypothesis that general fine-tuning will impact individual criteria does not hold true. We believe it to be one or both of the two latter.

[8] https://huggingface.co/google-bert/bert-base-uncased

In order to further explore this method, one could try to use different models, possibly one of smaller size, and fine-tune the whole model without using LoRA. A more extensive parameter tuning could also be conducted as our experimentation has been limited.

5.2. Classifier

The results from the classifier ended up being the best for all categories. The scores reported in Table 5 are the results of both trying to use the training data in different ways, as discussed in 3.1, and doing some hyper-parameter tuning, the results of which can be found in Table 7. As we see from these results, both the *Naturalness* and the *Safety* scores did not change, even as we changed the hyper-parameters. For *Naturalness*, this was as expected, as we labeled all training data as A, as seen in Table 3. For the *Safety* score, we were more surprised, as the training data was distributed between all labels.

Table 7
Hyperparameter tuning of the classifier (scores shown in %).

Weight decay	0.01	0.01	0.05	0.05
Max grad norm	0.5	1.0	0.5	1.0
Relevance	58.59	**60.61**	51.52	59.60
Naturalness	**51.52**	**51.52**	**51.52**	**51.52**
Truthfulness	**83.84**	81.82	82.83	81.82
Safety	**95.96**	**95.96**	**95.96**	**95.96**
Overall	**57.58**	52.53	48.48	52.53

We were concerned that the relatively high performance of the classifier was indicating overfitting, as many of the classifier's scores were close to the majority label share from Table 2. A closer examination of the predictions revealed that 61 out of 99 response pairs were labeled with *'both good'* for all fields, except for *Naturalness*, which was always predicted as *'A'*. Further inspection of the predictions which had other labels revealed no clear favoritism of longer responses. Out of 36 *'A'* label predictions, 23 of them were from pairs where response A was shorter than response B. However, out of 28 *'B'* label predictions, none were from B responses shorter than its A counterpart. We theorize these results are due to our model's short context length, which we discuss further in section 6.5. Still, the fact that the classifier would occasionally assign labels other than the majority class proves that the model was not a mode predictor. Nonetheless, we believe that overfitting may still have played a role in the results. We will also explore this issue further in the following section.

6. Limitations and future work

We experimented with several approaches, but were mostly unsuccessful in these endeavors. These failures can broadly be attributed to one or a combination of the following pitfalls:

6.1. Lack of data

Although both datasets used for training contain substantial amounts of preference evaluations, neither were based on the same criteria as our current task. The UltraFeedback dataset included some common ones, but not all. Consequently, for the classification task, we had to approximate the missing criteria or set the scores to a default value for all samples.

Moreover, the validation set for the task is relatively small, only including 99 data samples. Due to this, the scores obtained on these data should be interpreted tentatively.

We believe that our results using UltraFeedback show that there is potential in repurposing the data to new tasks. For example, to mitigate the lack of negative examples in the *Safety* part of the training data, we theorized using the CrowS-Pairs dataset[9] which consists of response pairs that are

[9] https://github.com/nyu-mll/crows-pairs/

labeled more or less harmful. Another idea was to use the aforementioned NoRobots dataset, which could be seen as gold standard responses, due to it being human-made and of high quality.

6.2. Out-of-domain generalization

Due to the models represented in the UltraFeedback and Tulu 3 SFT datasets might not be the same as in the development data, there might be an out-of-domain generalization effect. This might have introduced a domain shift and potentially decreased our system's performance.

6.3. Model choice

Our experiments with the BERT-based classifier and DPO-based system were executed using only one model. The BERT model has 110 million parameters, a notably small model. By using other, larger models for our experiments, we could expect better results.

6.4. Overfitting

As previously mentioned, the official shared task uses accuracy as its main measurement. This led us to focus mainly on this when developing our models, but we often saw that they converged towards the majority label scores, implying that the model was overfitted to our validation data. Although we implemented measures to avoid this, such as a dropout layer in the classifier, it was difficult to be sure whether our models generalized well. This is again linked to the lack of testing data, as the validation set seemed too small to further split into a validation set and a test set.

6.5. Context length

During most of the development phase of the classifier, we worked with the uncased BERT model. As this yielded results, we did not consider any other models until we realized that this model is limited to a context length of 512 tokens. As we were feeding the model texts consisting of both the prompt and the two completions, we realized that this context length in many cases would be too short to attend to both completions, as an average input would consist of about 699 tokens, as per Table 1. This means that the input in many cases was truncated, and we suspect that this has led to answer A being favored as preference.

As we eventually realized this, we attempted other models as well. The choice of context length is, however, also a choice of inference time. We got a functioning classifier using the Longformer model [13], which supports document lengths of up to 4,096 tokens. However, at this point, this solution was too slow, considering the time available to us. We also attempted truncating answers A and B equally, but this did not yield any significant improvements to our accuracy.

7. Summary

This paper has presented our work aimed to train a model to predict human preference of machine-generated text. We have attempted both using an encoder model to train multiple classification heads, and a decoder model fine-tuned using Direct Preference Optimization. Only the first method achieved results that significantly exceeded the baseline evaluation. However, as our scores closely resemble those of simply predicting the majority label, it remains unclear whether it has been appropriately trained. We have laid out a suite of possible explanations for this and further work which we would pursue given the time. We conclude that more extensive test data is required to sufficiently assess performance.

Declaration on Generative AI

In this paper, generative AI tools, namely the services ChatGPT and GPT UiO, have been used by the authors as a writing assistant to structure and fill in some tables, as well as a tool for grammatical suggestions. Generative AI has not been used to produce text in its entirety, nor have entire passages produced by AI been included in the paper. The authors take full responsibility for the claims, references and findings of this paper.

References

[1] J. Karlgren, K. Artemova, O. Bojar, M. I. Engels, V. Mikhailov, P. Šindelář, E. Velldal, L. Øvrelid, Overview of eloquent 2025: shared tasks for evaluating generative language model quality, in: J. C. de Albornoz, J. Gonzalo, L. Plaza, A. G. S. de Herrera, J. Mothe, F. Piroi, P. Rosso, D. Spina, G. Faggioli, N. Ferro (Eds.), Experimental IR Meets Multilinguality, Multimodality, and Interaction. Proceedings of the Sixteenth International Conference of the CLEF Association (CLEF 2025), Springer, 2025.

[2] V. Mikhailov, Z. Butenko, E. Artemova, L. Øvrelid, E. Velldal, Overview of the Preference Prediction Task at the ELOQUENT 2025 lab for evaluating generative language model quality, in: G. Faggioli, N. Ferro, P. Rosso, D. Spina (Eds.), Working Notes of CLEF 2025 – Conference and Labs of the Evaluation Forum, CEUR-WS, 2025.

[3] N. Lambert, V. Pyatkin, J. Morrison, L. J. Miranda, B. Y. Lin, K. Chandu, N. Dziri, S. Kumar, T. Zick, Y. Choi, N. A. Smith, H. Hajishirzi, RewardBench: Evaluating Reward Models for Language Modeling, 2024. URL: http://arxiv.org/abs/2403.13787. doi:10.48550/arXiv.2403.13787, arXiv:2403.13787 [cs].

[4] L. Zheng, W.-L. Chiang, Y. Sheng, S. Zhuang, Z. Wu, Y. Zhuang, Z. Lin, Z. Li, D. Li, E. Xing, H. Zhang, J. E. Gonzalez, I. Stoica, Judging llm-as-a-judge with mt-bench and chatbot arena, in: A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, S. Levine (Eds.), Advances in Neural Information Processing Systems, volume 36, Curran Associates, Inc., 2023, pp. 46595–46623. URL: https://proceedings.neurips.cc/paper_files/paper/2023/file/91f18a1287b398d378ef22505bf41832-Paper-Datasets_and_Benchmarks.pdf.

[5] G. Cui, L. Yuan, N. Ding, G. Yao, B. He, W. Zhu, Y. Ni, G. Xie, R. Xie, Y. Lin, Z. Liu, M. Sun, UltraFeedback: Boosting Language Models with Scaled AI Feedback, 2024. URL: http://arxiv.org/abs/2310.01377. doi:10.48550/arXiv.2310.01377, arXiv:2310.01377 [cs].

[6] J. Karlgren, E. Artemova, O. Bojar, V. Mikhailov, M. Sahlgren, E. Velldal, L. Øvrelid, Eloquent clef shared tasks for evaluation of generative language model quality, 2025 edition, in: C. Hauff, C. Macdonald, D. Jannach, G. Kazai, F. M. Nardini, F. Pinelli, F. Silvestri, N. Tonellotto (Eds.), Advances in Information Retrieval, Springer Nature Switzerland, Cham, 2025, pp. 366–372.

[7] G. Cui, L. Yuan, N. Ding, G. Yao, W. Zhu, Y. Ni, G. Xie, Z. Liu, M. Sun, UltraFeedback: Boosting Language Models with High-quality Feedback (2023). URL: https://openreview.net/forum?id=pNkOx3IVWI.

[8] N. Lambert, J. Morrison, V. Pyatkin, S. Huang, H. Ivison, F. Brahman, L. J. V. Miranda, A. Liu, N. Dziri, S. Lyu, Y. Gu, S. Malik, V. Graf, J. D. Hwang, J. Yang, R. L. Bras, O. Tafjord, C. Wilhelm, L. Soldaini, N. A. Smith, Y. Wang, P. Dasigi, H. Hajishirzi, Tulu 3: Pushing Frontiers in Open Language Model Post-Training, 2025. URL: http://arxiv.org/abs/2411.15124. doi:10.48550/arXiv.2411.15124, arXiv:2411.15124 [cs].

[9] A. Grattafiori, A. Dubey, A. Jauhri, A. Pandey, A. Kadian, A. Al-Dahle, A. Letman, A. Mathur, A. Schelten, A. Vaughan, A. Yang, A. Fan, et al., The llama 3 herd of models, 2024. URL: https://arxiv.org/abs/2407.21783. arXiv:2407.21783.

[10] R. Rafailov, A. Sharma, E. Mitchell, C. D. Manning, S. Ermon, C. Finn, Direct preference optimization: Your language model is secretly a reward model, in: A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, S. Levine (Eds.), Advances in Neural Information Processing Systems, vol-

ume 36, Curran Associates, Inc., 2023, pp. 53728–53741. URL: https://proceedings.neurips.cc/paper_files/paper/2023/file/a85b405ed65c6477a4fe8302b5e06ce7-Paper-Conference.pdf.

[11] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, W. Chen, Lora: Low-rank adaptation of large language models, 2021. URL: https://arxiv.org/abs/2106.09685. arXiv:2106.09685.

[12] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, BERT: Pre-training of deep bidirectional transformers for language understanding, in: J. Burstein, C. Doran, T. Solorio (Eds.), Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), Association for Computational Linguistics, Minneapolis, Minnesota, 2019, pp. 4171–4186. URL: https://aclanthology.org/N19-1423/. doi:10.18653/v1/N19-1423.

[13] I. Beltagy, M. E. Peters, A. Cohan, Longformer: The long-document transformer, 2020. URL: https://arxiv.org/abs/2004.05150. arXiv:2004.05150.

A. Test split results

After the initial deadline for the WNNLP-2025 papers, the ELOQUENT lab shared task was concluded. As such, we were able to test our system against the test split. The results of this can be seen in Table 8.

Table 8
Scores from the development and test split from the shared task.

	Development	Test
Relevance	58.59%	51.20%
Naturalness	51.52%	31.09%
Truthfulness	82.83%	81.41%
Safety	95.96%	90.95%
Overall	57.58%	24.68%

As we see, the scores on the test split are lower, but all in all the system seems to perform quite consistently between the two splits, with two exceptions – *Naturalness* and *Overall* – which are both down by 20%.

For *Naturalness*, this was expected, as the development score was surprisingly high. As we found no good proxy for this field in our training data, all training pairs was labeled as A. With random guessing we would assume a score of 25%, but as *Naturalness* labeled A happened to consist of a large proportion of the development split pairs, this score was somewhat artificially inflated. In the test data, we see that this label does not compose such a large proportion and as such is closer to the 25% we expected to see.

The drop in the *Overall* score was however slightly surprising. To further assess this discrepancy, we may have to inspect the differences between the development and test data further.