Using LLMs and Preference Optimization for Agreement-Aware HateWiC Classification

Sebastian Loftus[*2], **Adrian Mülthaler**[*2], **Sanne Hoeken**[*1]
Sina Zarrieß[1] and **Özge Alaçam**[1,2]
[1]Computational Linguistics, Department of Linguistics, Bielefeld University, Germany
[2]Center for Information and Language Processing, LMU Munich, Germany
{s.loftus, adrian.muelthaler}@campus.lmu.de
{sanne.hoeken, sina.zarriess, oezge.alacam}@uni-bielefeld.de

Abstract

Annotator disagreement poses a significant challenge in subjective tasks like hate speech detection. In this paper, we introduce a novel variant of the HateWiC task that explicitly models annotator agreement by estimating the proportion of annotators who classify the meaning of a term as hateful. To tackle this challenge, we explore the use of Llama 3 models fine-tuned through Direct Preference Optimization (DPO). Our experiments show that while LLMs perform well for majority-based hate classification, they struggle with the more complex agreement-aware task. DPO fine-tuning offers improvements, particularly when applied to instruction-tuned models. Our results emphasize the need for improved modeling of subjectivity in hate classification and this study can serve as foundation for future advancements.

1 Introduction

Classification tasks involving subjective human judgment often exhibit annotator disagreement. This issue is particularly evident in hate speech detection, where the perception of hatefulness varies depending on context and individual interpretation (Yu et al., 2022). Ignoring disagreement in annotations can lead to biased systems that fail to account for minority perspectives (Davidson et al., 2019; Sap et al., 2022). Addressing this variability requires models to go beyond binary classification and account for the degree of disagreement among annotators (Fleisig et al., 2023).

One task that exemplifies this challenge is Hateful Word in Context (HateWiC) Classification, which determines whether the meaning of a given term is hateful within a specific context (Hoeken et al., 2024). The initial work introducing HateWiC explored several BERT-based embedding learning strategies, demonstrating that incorporating additional input information, such as word definitions and annotator-specific data, can enhance performance. In particular, the inclusion of annotator information appears promising given the subjective nature of the task.

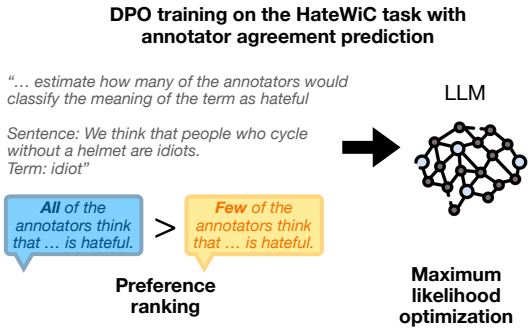

Figure 1: Illustration of our proposed approach to the HateWiC task, leveraging preference optimization via DPO training to predict annotator agreement.

In this paper, we propose a novel approach to the HateWiC task, leveraging Large Language Models (LLMs) fine-tuned via preference optimization. Recent advancements in preference-based learning, particularly Reinforcement Learning from Human Feedback (RLHF) (Ouyang et al., 2022) and its more computationally efficient alternatives like Direct Preference Optimization (DPO) (Rafailov et al., 2024), have driven significant improvements in LLM alignment for tasks such as question-answering and summarization (Nakano et al., 2021; Stiennon et al., 2020). However, their potential in subjective hate classification, where human disagreement plays a crucial role, remains unexplored.

We focus on two HateWiC task variants: 1) hate classification, where the goal is to predict the majority annotator judgment on whether a term is hateful and 2) **agreement-aware hate classification**, a novel task variant that we introduce to explicitly model annotator disagreement by estimating the proportion of annotators who classify a term as

[*]These authors contributed equally to this work.

Proceedings of the The 9th Workshop on Online Abuse and Harms (WOAH), pages 538–547

hateful. This task merges classification and annotator disagreement, capturing the subjective nature of hatefulness.

Contributions We introduce a preference-based fine-tuning approach for HateWiC, leveraging Direct Preference Optimization (DPO) to align LLMs with human judgments of hatefulness. We propose a novel variant of hate classification that integrates annotator disagreement directly into the learning objective, as illustrated in Figure 1. Evaluating both task variants, we compare the effect of DPO-tuning on models that are instruction tuned or not. Our results show that models without preference tuning perform effectively for general (majority-based) hate classification, and additional preference optimization does not yield further improvements. However, for agreement-aware hate classification, DPO fine-tuning enhances performance when applied to instruction-tuned models. Despite these improvements, the second task remains challenging for LLMs. These findings highlight both the potential and limitations of our approach, emphasizing the need for further research to better model annotator disagreement.[1]

2 Related Work

2.1 Preference optimization

Preference optimization trains models based on the principle that, given an input text, one response is preferred over another. This approach has proven effective in better aligning LLMs with human preferences (OpenAI et al., 2024; Tunstall et al., 2023). Most research in this area has focused on enhancing fluency and safety in general generation tasks like summarization and dialogue (Ziegler et al., 2020; Stiennon et al., 2020). However, less attention has been given to more specialized classification tasks, particularly in the context of diverse human preferences (Cheng et al., 2023).

The initial method for optimizing responses, RLHF with Proximal Policy Optimization (PPO) (Schulman et al., 2017) is computationally expensive, especially with massive models like Llama 3 (Grattafiori et al., 2024). Recently, new algorithms have emerged to streamline RLHF, reduce training costs, and improve efficiency. Direct Preference Optimization (DPO) (Rafailov et al., 2024) merges the reward model training and RLHF training into a single step by combining supervised fine-tuning (SFT) on positive samples with reverse SFT on negative samples.

2.2 Hate speech detection

Hate speech detection (HSD) has been extensively studied using various approaches, particularly with transformer-based language models. Early methods fine-tuned encoder-based transformers such as BERT for classification (Sarkar et al., 2021; Caselli et al., 2021). More recently, prompt-based approaches leveraging LLMs have gained attention, demonstrating strong zero- and few-shot capabilities for HSD, especially with instruction-tuned models (Chiu and Alexander, 2021; Plaza-del arco et al., 2023; Ronghao Pan, 2024).

Some preference optimization methods have been applied to related tasks such as sexism detection (Riahi Samani et al., 2025) and counter speech generation (Wadhwa et al., 2025). However, many common HSD approaches, including the aforementioned, often overlook the inherent subjectivity of hate speech annotation, which has been increasingly recognized as an important challenge. Prior work has explored modeling annotator disagreement rather than relying solely on majority voting (Mostafazadeh Davani et al., 2022; Wan et al., 2023). Yet, existing approaches do not leverage preference optimization to align models with human judgments on *subjective* hatefulness.

3 Tasks & Data

In this section, we briefly describe the HateWiC dataset and the tasks addressed in this study, outlining how we create preference pairs for DPO training.

3.1 HateWiC dataset

The HateWiC dataset (Hoeken et al., 2024) is a dataset comprising approximately 4,000 instances of (non-)hateful terms in example sentences, scraped from Wiktionary. Each instance in the dataset is annotated by three individuals, who provide labels indicating the perceived hatefulness of the term within its specific context. The dataset includes both the majority label and individual annotator labels; we use the former for Task 1 and the latter for Task 2, as described below.

3.2 Task 1 - Hate Classification

For the task of hate classification, we construct preference pairs for DPO training using the HateWiC

[1]The code used for this study can be found at: https://github.com/sebloft/DPO4AgreeAwareHateWiC

dataset. The goal is to train a model to classify whether the meaning of a given term within a specific sentence is hateful. The input prompt, as provided below, instructs the model to determine whether a term in a given sentence is *hateful* or *not hateful*.

> ### Instruction:
> Given the following sentence that mentions a particular term, classify whether the meaning of that term expresses hate towards a person or group within that specific sentence by giving one of the following corresponding labels:
> "hateful"
> "not hateful"
>
> ### Input:
> Sentence: [SENTENCE]
> Term: [TERM]
>
> ### Response:

To facilitate preference-based learning, we construct pairwise preference outputs by generating a positive and a negative response, where the positive response aligns with the majority binary hatefulness label, while the negative response provides the incorrect classification. The specific responses are formulated as follows:

- Positive: "The meaning of [TERM] in the text [SENTENCE] is [CORRECT HATE LABEL]"
- Negative: "The meaning of [TERM] in the text [SENTENCE] is [INCORRECT HATE LABEL]"

3.3 Task 2 - Agreement-Aware Hate Classification

Agreement-aware hate classification estimates the distribution of human annotator judgments by predicting the proportion of annotators who classify a (contextualized) term as hateful using predefined categories: *all*, *most*, *half*, *few*, or *none*. With the input prompt being formulated as:

> ### Instruction:
> Given the following sentence that mentions a particular term, estimate how many of the human annotators would classify the meaning of that term as hateful by giving one of the following quantifiers:
> "all"
> "most"
> "half"
> "few"
> "none"
>
> ### Input:
> Sentence: [SENTENCE]
> Term: [TERM]
>
> ### Response:

To assess the robustness of our approach, we additionally test alternative prompt formulations, which are reported in Appendix C.

For DPO training, we construct pairwise preference outputs where the positive response selects the correct quantifier aligned with the human annotation distribution while the negative response selects an incorrect quantifier (see also Figure 1):

- Positive: "[CORRECT QUANTIFIER] of the annotators think that the meaning of [TERM] in the text [SENTENCE] is hateful"
- Negative: "[INCORRECT QUANTIFIER] of the annotators think that the meaning of [TERM] in the text [SENTENCE] is hateful"

To select the correct quantifier, we consider the number of annotators who classify the instance as hateful out of the total number of annotations, typically three. For example, if two out of three annotators classify a term as hateful, the quantifier *most* is chosen, while if only one annotator marks it as hateful, the quantifier *few* is selected. To ensure a clear contrast with the negative response, we use fixed mappings from correct to incorrect quantifiers, avoiding hierarchical overlap (e.g., preventing *all* from being replaced with *most*, as *all* inherently includes *most*).

4 Methods

This section details the experimental set-up for our experiments, including the DPO training paradigm, the chosen models and the evaluation pipeline.

4.1 Models and Training

We use two distinct 8B-sized Llama 3 model checkpoints, each developed using a different post-training paradigm after pre-training. The first,

which we refer to as **Sft**, is a Supervised Fine-Tuned (SFT) model[2]. The second, referred to as **Instruct**, was trained with SFT followed by preference tuning via RLHF[3]. Unlike Sft, the **Instruct** model was further optimized to align more closely with human values using human-annotated preference data[4]. For each of these models, we fine-tune them on HateWiC data using DPO on two tasks, resulting in two further variants per task: **Sft-tuned** and **Instruct-tuned** models. Due to compute limitations, 4-bit quantization was applied before training and evaluation using the bitsandbytes library[5] and peft (Mangrulkar et al., 2022) was used for more efficient fine-tuning. For training, the trl package[6] was used, which provides an extensive preference optimization framework (von Werra et al., 2020). Details on the hardware and the training setup can be found in Appendix B.

4.2 Evaluation setup

We employed a ten-fold cross validation setup, using for each run eight folds for training (approx. 3100 instances), one for development, and one for testing (approx. 390 instances).

For the evaluation of hate classification (Task 1), we extracted the binary labels, *hateful* and *not hateful*, from the model outputs using pattern matching. Instances without a valid generated label were excluded from the evaluation. In the agreement-aware hate classification (Task 2), we compared the predicted distribution of *hateful* annotations with the real human label distribution, both expressed using natural language quantifiers (as explained in 3.3). Again, labels were extracted through pattern matching, and instances without valid generated labels were omitted.

For both tasks, we report average F1 and Accuracy scores across all three folds for each fine-tuned model. This provides a comparative analysis of the performance between Sft and Instruct models, both with and without additional preference optimization. Additionally, we compare the models against a majority-vote baseline.

5 Results & Discussion

This section presents the results of our methods on two variants of the HateWiC task.

[2]huggingface.co/OpenRLHF/Llama-3-8b-sft-mixture
[3]huggingface.co/OpenRLHF/Llama-3-8b-rlhf-100k
[4]For details on the models, see Dong et al. (2024)
[5]pypi.org/project/bitsandbytes
[6]https://pypi.org/project/trl

Task 1	Hate	No Hate	Acc.	Macro
Sft	**0.755**	0.765	0.761	0.760
Sft-tuned	0.751	0.774	**0.763**	**0.762**
Instruct	0.517	0.763	0.675	0.640
Instruct-tuned	0.602	**0.777**	0.708	0.689
N	1815	2030	3845	3845

Table 1: F1-scores of our four models on Task 1 of hate classification for both Hate and No Hate classes, as well as Accuracy and Macro F1 for overall performance.

Sft-only suffices for majority-based hate classification. Table 1 presents the performance results of four models on Task 1. Overall, the Sft models achieve the best performance with a macro F1 score of 0.77. The effect of DPO fine-tuning on the HateWiC data appears negligible. Notably, the Instruct models underperform compared to the Sft models, particularly on the *hate* class (0.53 F1). These results suggest that (1) instruction tuning may make the model more conservative in predicting hate speech and (2) general pre-training of Llama 3 (with SFT) already provides sufficient knowledge for detecting hate speech at a broad level, aligning with majority judgments.

Moreover, our Sft-models are competitive with the best BERT-based approach as reported in the original HateWiC paper (Hoeken et al., 2024) (0.78 accuracy). The authors also reported that zero-shot Llama 2 performed worse (0.68 accuracy). Our results align with their conclusion that, despite their strong performance elsewhere, Llama models do not demonstrate superior performance over BERT-based methods on this task.

Task 2	All	Most	Few	None	Acc.	Macro
BL - Majority	0.000	0.000	0.000	0.496	0.330	0.124
Sft	0.031	0.275	0.150	0.627	**0.390**	0.271
Sft-tuned	0.066	0.211	0.093	0.591	0.376	0.240
Instruct	0.040	0.302	**0.242**	**0.647**	0.382	0.308
Instruct-tuned	**0.071**	**0.324**	0.205	0.641	0.387	**0.311**
N	971	844	761	1269	3845	3845

Table 2: F1-scores of our four models on Task 2 of agreement-aware hate classification for each of the four classes, as well as Macro F1 and Accuracy for overall performance.

DPO enhances agree-aware hate classification, but the task remains challenging. As can be seen in Table 2, the performance results shift when evaluating Task 2, which explicitly incorporates subjectivity and annotator (dis)agreement. This additional complexity makes the task notably more difficult for LLMs, as reflected in significantly

lower performance compared to Task 1. While the models improve upon the majority-voting baseline, the improvement is modest. Considering the macro F1 scores, which address class imbalance, the best performance is achieved by applying DPO fine-tuning to the Instruct model.

When examining class-wise performance, all models struggle most with the *all* category, followed by *few* and *most*, while the *none* category yields the best results. This pattern suggests that LLMs find it easier to align with clear-cut non-hate cases but struggle when the input is hateful or ambiguous, thus prioritizing caution over recall, potentially due to the challenges of handling subjectivity and disagreement between annotators. Appendix D provides a more detailed error analysis.

Instruct-models can benefit from task-specific preference tuning, Sft-models not. In Task 1 we observe that task-specific DPO fine-tuning has minimal impact on the Sft model, but it substantially improves the performance of the Instruct model. Similarly, in Task 2, the effect of DPO fine-tuning varies between the two base models: it improves performance for the Instruct model, while it degrades the performance of the Sft model. These results suggest that for these tasks, Sft models appear less flexible to incorporating task-specific preference signals whereas instruction-tuned models benefit from such additional preference fine-tuning.

6 Conclusion

This paper addresses the challenges of incorporating subjective human judgment, particularly annotator disagreement, in tasks like HateWiC classification. We introduce a novel variant of the task, agreement-aware hate classification, which explicitly models the variability in human judgments. To tackle this task, we explore approaches using LLMs with DPO. Our findings show that pre-trained LLMs perform effectively for majority-based hate classification. However, these models struggle with the added complexity of agreement-aware hate classification. While DPO fine-tuning shows promise in enhancing performance, particularly when applied to instruction-tuned models, our study also emphasizes that further research is needed to better capture the subjective nature of hate speech detection. The novel task we present could serve as valuable foundation for future efforts.

Limitations

While our findings provide valuable insights, they are subject to several limitations. Due to hardware constraints, we relied on smaller 4-bit quantized models. Running our experiments on larger models could provide a more comprehensive evaluation of the effectiveness of our proposed method. Additionally, the computational demands of training LLMs necessitated certain trade-offs, particularly in optimizing all components of the training pipeline, such as hyperparameter tuning. Given these constraints, we prioritized methodological robustness by conducting evaluations across ten independent runs. Future research could enhance the reliability and generalizability of our findings by systematically exploring a broader range of hyperparameter settings, and assessing performance on larger-scale models.

Ethics Statement

Hate speech is a sensitive domain, and the reproduction of certain terms may be distressing to some readers. To promote fairness, we report our findings without explicitly using hateful terms. Moreover, we model annotator disagreement to account for minority perspectives rather than relying solely on majority votes. By incorporating agreement-aware classification, we aim to foster a more inclusive understanding of harmful language.

Our study makes use of an existing dataset that comprises annotations on hate speech, which includes annotator information. However, we do not utilize any personally identifiable information, ensuring the privacy of all annotators. We also ensure that our dataset usage aligns with its intended use.

Lastly, training LLMs is computationally expensive, contributing to a significant carbon footprint. To address this, we employ quantization techniques for more efficient model training.

Acknowledgements

The authors acknowledge financial support by the project "SAIL: SustAInable Life-cycle of Intelligent Socio-Technical Systems" (Grant ID NW21-059A), which is funded by the program "Netzwerke 2021" of the Ministry of Culture and Science of the State of North Rhine-Westphalia, Germany.

In addition, the authors acknowledge the use of an AI-based language assistant to refine wording and improve the readability of certain sections of

this paper. No AI-generated content was used for conceptual development.

References

Tommaso Caselli, Valerio Basile, Jelena Mitrović, and Michael Granitzer. 2021. HateBERT: Retraining BERT for abusive language detection in English. In *Proceedings of the 5th Workshop on Online Abuse and Harms (WOAH 2021)*, pages 17–25, Online. Association for Computational Linguistics.

Pengyu Cheng, Jiawen Xie, Ke Bai, Yong Dai, and Nan Du. 2023. Everyone deserves a reward: Learning customized human preferences. *Preprint*, arXiv:2309.03126.

Ke-Li Chiu and Rohan Alexander. 2021. Detecting hate speech with GPT-3. *CoRR*, abs/2103.12407.

Thomas Davidson, Debasmita Bhattacharya, and Ingmar Weber. 2019. Racial bias in hate speech and abusive language detection datasets. In *Proceedings of the Third Workshop on Abusive Language Online*, pages 25–35, Florence, Italy. Association for Computational Linguistics.

Hanze Dong, Wei Xiong, Bo Pang, Haoxiang Wang, Han Zhao, Yingbo Zhou, Nan Jiang, Doyen Sahoo, Caiming Xiong, and Tong Zhang. 2024. Rlhf workflow: From reward modeling to online rlhf. *Preprint*, arXiv:2405.07863.

Eve Fleisig, Rediet Abebe, and Dan Klein. 2023. When the majority is wrong: Modeling annotator disagreement for subjective tasks. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 6715–6726, Singapore. Association for Computational Linguistics.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, and 542 others. 2024. The llama 3 herd of models. *Preprint*, arXiv:2407.21783.

Sanne Hoeken, Sina Zarrieß, and Özge Alacam. 2024. Hateful word in context classification. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 172–186, Miami, Florida, USA. Association for Computational Linguistics.

Sourab Mangrulkar, Sylvain Gugger, Lysandre Debut, Younes Belkada, Sayak Paul, and Benjamin Bossan. 2022. Peft: State-of-the-art parameter-efficient fine-tuning methods. `https://github.com/huggingface/peft`.

Aida Mostafazadeh Davani, Mark Díaz, and Vinodkumar Prabhakaran. 2022. Dealing with disagreements: Looking beyond the majority vote in subjective annotations. *Transactions of the Association for Computational Linguistics*, 10:92–110.

Reiichiro Nakano, Jacob Hilton, Suchir Balaji, Jeff Wu, Long Ouyang, Christina Kim, Christopher Hesse, Shantanu Jain, Vineet Kosaraju, William Saunders, Xu Jiang, Karl Cobbe, Tyna Eloundou, Gretchen Krueger, Kevin Button, Matthew Knight, Benjamin Chess, and John Schulman. 2021. Webgpt: Browser-assisted question-answering with human feedback. *CoRR*, abs/2112.09332.

OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mohammad Bavarian, Jeff Belgum, and 262 others. 2024. Gpt-4 technical report. *Preprint*, arXiv:2303.08774.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, NIPS '22, Red Hook, NY, USA. Curran Associates Inc.

Flor Miriam Plaza-del arco, Debora Nozza, and Dirk Hovy. 2023. Respectful or toxic? using zero-shot learning with language models to detect hate speech. In *The 7th Workshop on Online Abuse and Harms (WOAH)*, pages 60–68, Toronto, Canada. Association for Computational Linguistics.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D. Manning, and Chelsea Finn. 2024. Direct preference optimization: Your language model is secretly a reward model. *Preprint*, arXiv:2305.18290.

Ali Riahi Samani, Tianhao Wang, Kangshuo Li, and Feng Chen. 2025. Large language models with reinforcement learning from human feedback approach for enhancing explainable sexism detection. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 6230–6243, Abu Dhabi, UAE. Association for Computational Linguistics.

Rafael Valencia-García Ronghao Pan, José Antonio García-Díaz. 2024. Comparing fine-tuning, zero and few-shot strategies with large language models in hate speech detection in english. *Computer Modeling in Engineering & Sciences*, 140(3):2849–2868.

Maarten Sap, Swabha Swayamdipta, Laura Vianna, Xuhui Zhou, Yejin Choi, and Noah A. Smith. 2022. Annotators with attitudes: How annotator beliefs

and identities bias toxic language detection. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5884–5906, Seattle, United States. Association for Computational Linguistics.

Diptanu Sarkar, Marcos Zampieri, Tharindu Ranasinghe, and Alexander Ororbia. 2021. fBERT: A neural transformer for identifying offensive content. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 1792–1798, Punta Cana, Dominican Republic. Association for Computational Linguistics.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal policy optimization algorithms. *CoRR*, abs/1707.06347.

Melanie Sclar, Yejin Choi, Yulia Tsvetkov, and Alane Suhr. 2024. Quantifying language models' sensitivity to spurious features in prompt design or: How i learned to start worrying about prompt formatting. *Preprint*, arXiv:2310.11324.

Nisan Stiennon, Long Ouyang, Jeffrey Wu, Daniel Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul F Christiano. 2020. Learning to summarize with human feedback. In *Advances in Neural Information Processing Systems*, volume 33, pages 3008–3021. Curran Associates, Inc.

Lewis Tunstall, Edward Beeching, Nathan Lambert, Nazneen Rajani, Kashif Rasul, Younes Belkada, Shengyi Huang, Leandro von Werra, Clémentine Fourrier, Nathan Habib, Nathan Sarrazin, Omar Sanseviero, Alexander M. Rush, and Thomas Wolf. 2023. Zephyr: Direct distillation of lm alignment. *Preprint*, arXiv:2310.16944.

Leandro von Werra, Younes Belkada, Lewis Tunstall, Edward Beeching, Tristan Thrush, Nathan Lambert, Shengyi Huang, Kashif Rasul, and Quentin Gallouédec. 2020. Trl: Transformer reinforcement learning. `https://github.com/huggingface/trl`.

Sahil Wadhwa, Chengtian Xu, Haoming Chen, Aakash Mahalingam, Akankshya Kar, and Divya Chaudhary. 2025. Northeastern uni at multilingual counterspeech generation: Enhancing counter speech generation with LLM alignment through direct preference optimization. In *Proceedings of the First Workshop on Multilingual Counterspeech Generation*, pages 19–28, Abu Dhabi, UAE. Association for Computational Linguistics.

Ruyuan Wan, Jaehyung Kim, and Dongyeop Kang. 2023. Everyone's voice matters: Quantifying annotation disagreement using demographic information. *Proceedings of the AAAI Conference on Artificial Intelligence*, 37(12):14523–14530.

Xinchen Yu, Eduardo Blanco, and Lingzi Hong. 2022. Hate speech and counter speech detection: Conversational context does matter. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5918–5930, Seattle, United States. Association for Computational Linguistics.

Daniel M. Ziegler, Nisan Stiennon, Jeffrey Wu, Tom B. Brown, Alec Radford, Dario Amodei, Paul Christiano, and Geoffrey Irving. 2020. Fine-tuning language models from human preferences. *Preprint*, arXiv:1909.08593.

A Data

We retrieved the HateWiC dataset upon request which is available for research purposes, licensed under CC BY-NC 4.0.

B Training specifications

Training was performed on a NVIDIA RTX A6000 (48 GB) a learning rate of 5e-7 and batch size of 4. All other hyperparameters (optimizer, reference model update strategy) are set to the default values of the trl package (version 0.14.0). For quantization, PEFT (Mangrulkar et al., 2022) was used with a rank of 32 and an alpha value of 16. For each of the ten folds **`Sft`** was tuned for 5.5 hours for 15 epochs and **`Instruct`** for 3 hours for 10 epochs, resulting in a total training time of 85 hours. Evaluation for Task 1 took 4.43 hours and 8 hours for Task 2 resulting in a total evaluation time of 12.43 hours.

C Prompt Variations

Sclar et al. (2024) showed that the performance of LLMs of a given task is highly sensitive towards spurious details in the prompt design. To assess the robustness of our findings to changes in prompt formulation, we conducted a set of additional experiments on three prompt variants. These experiments were performed on a single fold (fold 1) of our ten-fold cross-validation setup, with results reported on the corresponding test set. Our goal was to explore whether the observed effects of DPO tuning persist across surface-level and semantic variations of the task prompt.

C.1 Variation 2A - Simpler language

This variation employed a version of the original prompt rewritten in simpler language, intended to test model robustness to stylistic changes without altering the task's core semantics.

Input prompt:

Instruction:
How many of the human annotators think that the term in the following sentence is hateful? Answer with one of the following quantifiers:
"all"
"most"
"half"
"few"
"none"

Input:
Sentence: [SENTENCE]
Term: [TERM]

Response:

Pairwise preference output:

- Positive: "[CORRECT QUANTIFIER] of the annotators think that [TERM] in the text [SENTENCE] is hateful"
- Negative: "[INCORRECT QUANTIFIER] of the annotators think that [TERM] in the text [SENTENCE] is hateful"

C.2 Variation 2B - Alternative quantifiers

To reduce potential confounding effects of variability in handling different types of quantifiers in agreement prediction, this variation replaced quantity-based terms with frequency-based quantifiers. The mapping of the original quantifiers of quantity to the alternative quantifiers of frequency is given in Table 3. Additionally, the preference output was reformulated to fit the alternative quantifiers.

Quantity	Frequency
all	always
most	often
half	sometimes
few	rarely
none	never

Table 3: Mapping of original quantifiers of quantity to quantifiers of frequency.

Instruction:
How many of the human annotators think that the term in the following sentence is hateful? Answer with one of the following quantifiers:
"always"
"often"
"sometimes"
"rarely"
"never"

Input:
Sentence: [SENTENCE]
Term: [TERM]

Response:

Pairwise preference output:

- Positive: "Annotators [CORRECT QUANTIFIER] think that the meaning of [TERM] in the text [SENTENCE] is hateful"
- Negative: "Annotators [INCORRECT QUANTIFIER] think that the meaning of [TERM] in the text [SENTENCE] is hateful"

for which the [QUANTIFIER] options are: "always", "often", "sometimes", "rarely" or "never".

C.3 Variation 2C - Simpler language & alternative quantifiers

This prompt combines the simplified linguistic style of Variation 2A with the frequency-based quantifiers introduced in 2B.

Instruction:
How often do human annotators think that the term in the following sentence is hateful? Answer with one of the following quantifiers:
"always"
"often"
"sometimes"
"rarely"
"never"

Input:
Sentence: [SENTENCE]
Term: [TERM]

Response:

Pairwise preference output:

- Positive: "Annotators [CORRECT QUANTIFIER] think that [TERM] in the text [SENTENCE] is hateful"

- Negative: "Annotators [INCORRECT QUANTIFIER] think that [TERM] in the text [SENTENCE] is hateful"

for which the [QUANTIFIER] options are: "always", "often", "sometimes", "rarely" or "never".

C.4 Results

Table 4 summarizes the results of our single-fold experiments across three prompt variants. Across all three prompt variations, performance was slightly lower than with the original prompt, supporting our interpretation that the task's difficulty stems more from its subjective nature rather than prompt formulation. However, the effects of DPO tuning varied.

In Variation 2A, neither DPO-tuned model outperforms the untuned Instruct model, whereas Variations 2B and 2C show clearer gains from DPO tuning, particularly for the SFT model in 2C and the Instruct model in 2B. This indicates that both SFT and Instruct models can benefit from preference optimization, though gains are contingent on prompt structure. These results also suggest that model robustness may be more sensitive to surface-level linguistic variation than to the semantic structure of prompts (e.g., the way quantification is framed).

	Task 2 (orig.)		Variation 2A		Variation 2B		Variation 2C	
	Acc	F1	Acc	F1	Acc	F1	Acc	F1
SFT	0.405	0.269	0.392	0.217	0.238	0.179	0.292	0.252
SFT-tuned	0.390	0.240	0.387	0.208	0.277	0.234	**0.323**	**0.276**
Instruct	0.377	0.293	**0.395**	**0.255**	**0.300**	0.282	0.236	0.238
Instruct-tuned	**0.408**	**0.321**	0.382	0.248	**0.300**	**0.283**	0.256	0.248

Table 4: Accuracy and Macro-F1 on Task 2 with alternative prompt variations for Fold 1, with best Accuracy and Macro-F1 score highlighted per variation.

D Error Analysis

Content warning! This section contains examples of offensive language used solely for illustrative purposes. We are mindful of the impact such content may have.

Figure 2 presents the confusion matrices for our four models evaluated on Task 2 (Agreement-Aware Hate Classification). The SFT model demonstrates strong performance on the *none* class but performs poorly on *all* and *few*, frequently misclassifying *all* as *most* or *few*. The SFT-tuned variant shows modest improvements across *all*, *most*, and *few*, while maintaining high accuracy on *none*. The Instruct model offers a more balanced performance across classes than the SFT variants, with higher

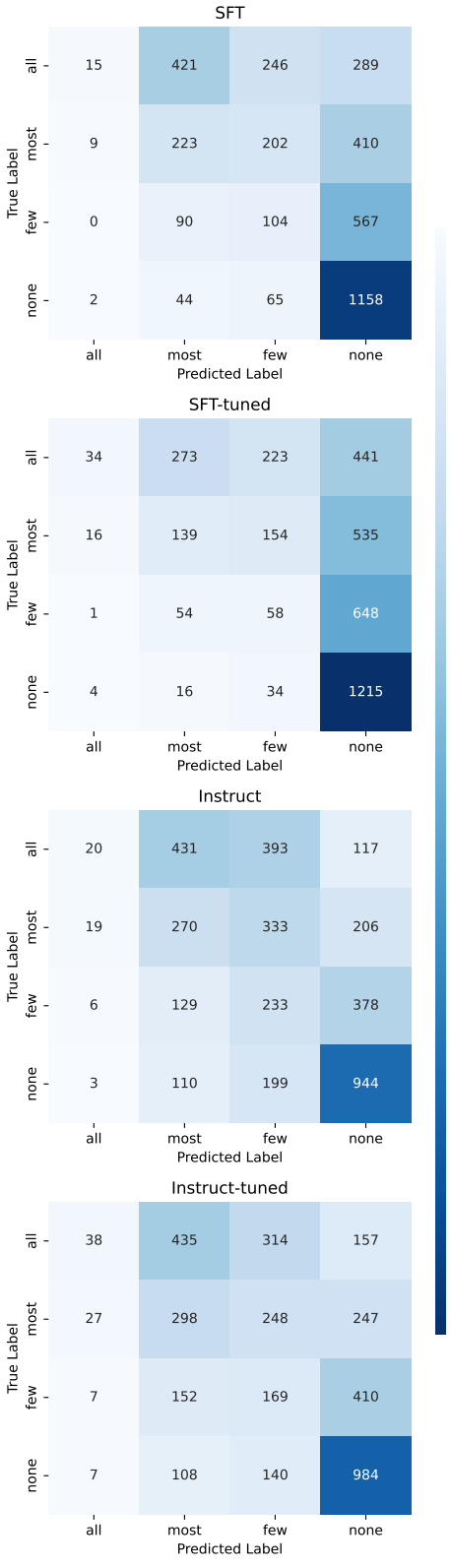

Figure 2: Confusion matrices for the four models evaluated on Task 2.

accuracy on *most* and *few*, though at the cost of reduced accuracy on *none*. The Instruct-tuned model achieves the best overall performance, with the highest accuracy on *all*, *most*, and *none*. While

Example	Term	Gold	SFT	SFT-tuned	Instruct	Instruct-tuned
She's a pure Oreo. You know, like the cookie, black outside and white inside.	Oreo	all	none	none	most	all
The village fool threw his own shoes down the well.	fool	most	none	none	none	most
Well, what do you think of the Canuck elections?	Canuck	few	none	none	few	few
How are you, you old bugger?	bugger	none	few	few	most	none

Table 5: Comparison of (erroneous) labels across models for various examples.

performance on *few* slightly declines compared to its untuned counterpart, it remains relatively strong. Across all models, there is a consistent tendency to overpredict the *none* class. However, the instruct-based models exhibit a more balanced distribution of predictions, suggesting greater sensitivity to class distinctions.

Table 5 presents some representative examples of model errors on Task 2. Each row compares human-based (Gold) labels with outputs from various model variants on selected HateWiC instances. As discussed, the models generally underperform relative to human annotations, but the Instruct-tuned model demonstrates relatively greater sensitivity in certain cases. For instance, in the "Oreo" example, where only the Instruct-tuned model aligned with the gold label, while other models failed to recognize the racially loaded meaning in context. Similarly, with the term "bugger", only Instruct-tuned captured its tone-dependent meaning, indicating a stronger grasp of pragmatic nuance.