Hate Speech Detection Project Report

Olga Ovcharenko
ETH Zürich
Zürich, Switzerland
oovcharenko@student.ethz.ch

Evžen Wybitul
ETH Zürich
Zürich, Switzerland
ewybitul@student.ethz.ch

Abstract

The rapid development of online media has given individuals a platform for expression and debate from the comfort of their homes. Yet, this freedom has inadvertently fueled a surge in online toxicity and hate speech. Social networks, forums, and chat room moderators are overwhelmed, as seen with Twitter, where users posted 500 million tweets per day in 2016 [22]. Current automatic detection methods fall short, particularly in pinpointing the targets of hate speech. The *Hate Speech Detection Project*[1] seeks to address this gap, providing a robust hate speech classification and target group detection models, which can be built upon to help Swiss and German newspaper moderators to combat hate speech, and to extract deeper insights from online data for research purposes.

1 Introduction

The problem of toxic speech detection has been researched for the past few decades. The challenges are both conceptual and practical; there is no single agreed-upon definition of "toxic speech", and toxicity is highly dependent on language and context. This complicates the construction of labeled datasets, making it a laborious task and consequently limiting the types of systems that can be trained for automatic hate speech detection.

Our project focuses on hate speech detection in user-generated comments on Swiss and German online media outlets. A large dataset of these comments has been collected and labeled by Kotarcic et al. [19], work we refer to as the *original paper* throughout this report. We build upon their dataset, presenting three models targeting specific subtasks within the general problem of harmful speech detection. Our contributions are:

- We document various problems and inconsistencies in the dataset, highlighting how its collection methodology limits the real-world applicability of models trained on it.
- We clean the dataset of these inconsistencies and prepare multiple evaluation sets to robustly gauge model performance in various contexts, which we hope will aid challenge givers in future model evaluations.
- We define three distinct tasks—toxicity detection, hate speech detection, and target group detection—and present a high-performing model for each. For the second task, our results surpass those reported in the original paper.
- Beyond research experiments, we emphasize practical model applications. We ensure full reproducibility of our data preprocessing pipeline, which we shared through a dedicated GitHub repository. We also make our top-performing models available on HuggingFace for easier access by challenge givers, and discuss how model training and inference performance depend on available computational resources.

In this report, we first define the three tasks, then discuss the original dataset and its limitations, followed by a description of our preprocessing pipeline setup and the construction of the evaluation sets. Subsequently, we address each task, detailing our methods and results. Finally, in the last two sections, we address overarching issues and propose directions for future work.

[1] Authors are sorted alphabetically.

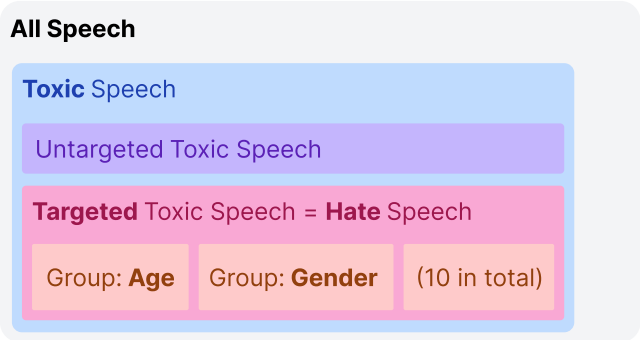

Figure 1: Relationships among the different types of speech

2 Background and Problem Statement

We define *toxic speech* as any form of communication that causes harm, aligning with existing definitions [4]. We distinguish between untargeted toxic speech and what is commonly referred to as *hate speech*, or targeted toxic speech. According to the United Nations, hate speech is communication that attacks or uses pejorative or discriminatory language against a person or group based on identity factors such as religion, ethnicity, nationality, or gender, among others[20]. Figure 1 illustrates the different terms and their interrelations. We adapt the set of target groups from the paper that informs our work to include: sexuality, age, gender, religion, nationality, political views, social status, disability, appearance, and the catch-all "other" covering for example cyberbullying or COVID-19 related discrimination.

Our project's goal is to create an economical and computationally efficient framework to process German comments from online media outlets. Based on the term hierarchy, we have identified three distinct tasks:

(1) *Toxic Speech Detection:* Determine if a comment contains toxic speech, either targeted or untargeted, as a binary classification task.
(2) *Hate Speech Detection:* Ascertain whether a comment includes targeted toxic speech, another binary classification task.
(3) *Hate Speech Target Group Detection:* Identify the group(s) targeted in a comment containing hate speech, a multilabel classification task, since comments may target multiple groups simultaneously.

The Methods and Results sections are structured to cover these three tasks separately.

3 Data

In this section, we first delve into the dataset compiled by Kotarcic et al. [19], which serves as the foundation for our project. We then outline our preprocessing steps, highlighting key deviations from the methods employed in the original study. Finally, we present our evaluation sets.

3.1 Original Dataset

The dataset was annotated by trained student research assistants following the definitions outlined in Section 2 and utilizing the scheme depicted in Figure 6. Comments identified as hate speech prompted annotators to further specify the targeted groups. Most comments were labeled by a single individual.

Composed of Swiss-German and French comments, the dataset also includes traces of other languages due to regional dialects and demographic diversity, as illustrated in Figure 2. To streamline the process, language detection was automated, although we acknowledge the potential for residual noise despite efforts to refine the detection methods.

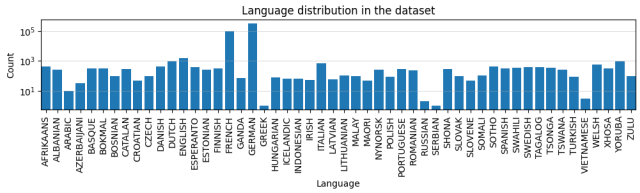

Figure 2: Language distribution in the dataset

Regarding label distribution, the dataset originally contained 422,000 comments, predominantly non-toxic and non-targeted (Figure 3, dataset ALL). Post-preprocessing, the dataset was reduced to approximately 200,000 comments. The distribution of target classes, as shown in Figure 5, reveals a skew towards certain categories, with others like appearance and disability being less represented. Moreover, Figure 4 indicates that comments often target 1 to 3 classes.

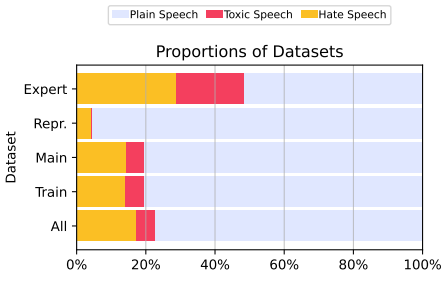

Figure 3: Label distribution in different subsets of our data

Sampling bias is a significant concern, as the dataset was compiled incrementally with subsequent batches of comments being

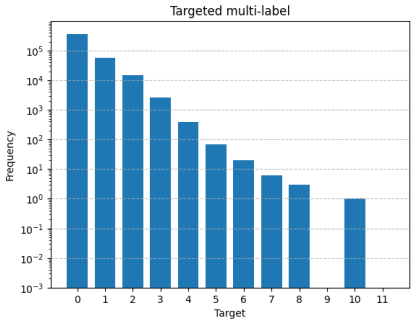

Figure 4: Target class cooccurrences

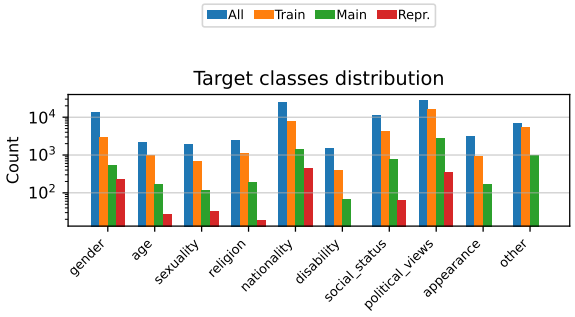

Figure 5: Target classes distribution

pre-selected by a model trained on previous data. This non-random selection resulted in an overrepresentation of toxic and hateful comments compared to what a true random sample would yield. To address this, we use the initial batch of 25,000 comments as a more representative test set, denoted as REPR. in Figure 3, contrasting it with the train (TRAIN) and main evaluation (MAIN) datasets, which reflect the selection bias. A smaller, expert-labeled batch (EXPERT) is also discussed later.

We encountered several data inconsistencies, such as duplicates, NaNs, and non-textual entries, as detailed in Table 1. Additionally, mismatches between labels and the definitions from Section 2 were evident (Table 2). For example, comments with similar semantic content received varying labels, indicating potential ambiguity in the operationalization of toxic speech concepts:

- *Junger unerfahrener mann 26jj. sucht lehrerin in konstanz* is labeled as untargeted and toxic, similarly
- *19jährigr sucht in böhlen (sachsen)* is labeled as untargeted and toxic, but
- *muskulöser & attraktiver m24 sucht hübsche sie (tg) in köln* is labeled is targeted but not toxic.

Quantifying these discrepancies is challenging, yet they underscore the complexity of consistently annotating such data.

3.2 Data Preprocessing

Our preprocessing approach for the comments mirrors that of the original study, with minimal alterations to the text. We remove

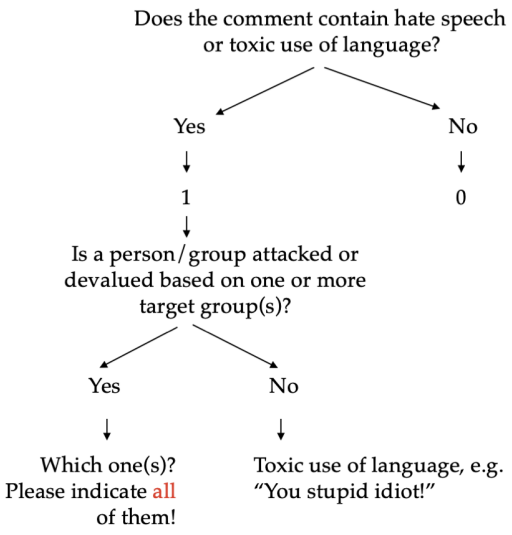

Figure 6: Annotation scheme from the original paper [19]

Table 1: Data Inconsistencies

Problem	# Occurrences
Duplicates	346
Not a Number	1237
Non-Textual Comment	13

Table 2: Label Inconsistencies

Problem	# Occurrences
Targeted w/o Target	804
Untargeted with Target	62
Targeted & Non-Toxic	13
Diverse Labels & Similar Context	No Data

any user-mentions, strip out URLs, convert emojis to their textual descriptions, standardize whitespace and newline characters, and excise HTML tags.

Contrary to the original study, we retain the original case of the text, recognizing its significance as an informational cue in German. We exclude comments with illogical label inconsistencies, eliminate all null values, and resolve duplicates by keeping a single instance of each comment, with its label determined by majority vote among the duplicates. Column names have been standardized to better reflect the definitions of toxic and hate speech. Language detection is performed using the *lingua* Python library. After consulting with our project partners, we decided to concentrate on German-language comments for our analyses.

A dedicated repository housing the complete preprocessing codebase has been established to facilitate reproducibility and future work by our collaborators [24].

3.3 Evaluation Sets

Initially, a single evaluation set which we split off from the main dataset was employed. However, upon understanding the dataset's sampling methodology, we integrated two additional sets to evaluate our model's generalizability:

(1) MAIN: Contains 15% of the dataset, collected via an active learning strategy. In collaboration with the challenge givers, we prioritized improving performance on this set.
(2) REPRESENTATIVE: Comprises the initial 25,000 comments, which were randomly sampled and serve as a more general representation.
(3) EXPERT: Includes 500 comments labeled through consensus among three experts.

Differences in label distributions across these sets are depicted in Figure 3 and are instrumental in understanding model performance variation.

The MAIN set, as well as validation and training sets, were prepared using an iterative stratification algorithm to maintain consistent label distribution across splits — this is reflected in the similar label distributions of the TRAIN and MAIN sets shown in Figure 3.

In addition to the EXPERT set, we have a comparable dataset labeled by five trained coders, referred to as CODERS throughout this text. These two datasets primarily serve to benchmark human performance against the automated classification tasks. We also use the EXPERT set to provide some basis for model-expert comparison.

In our efforts to address label imbalance in the training set, we initially considered SMOTE [5] as a solution. However, we ultimately chose not to use it. Studies have shown that SMOTE does not reliably improve performance on NLP tasks [14]. Moreover, our experiments revealed that the synthetic text generated through SMOTE's continuous interpolation lacked semantic coherence, resulting in nonsensical strings of mixed-language characters.

4 Methods

In this section, we outline the experiments conducted and the methodologies employed for each task. Our experiments were logged in a shared Weights & Biases workspace [2] to promote reproducibility and provide challenge givers with direct access to the results.

4.1 Toxic speech detection

Our objective was to replicate and surpass the results of the original paper, comparing diverse methods to identify the most suitable for the problem.

Baselines: For our first baseline, we used a *Constant baseline* predictor that classifies all entries as toxic speech; this baseline was expected to yield the highest F1 and AUPRC scores among trivial baselines.

The second baseline employed was FastText [3], a library developed by Facebook for text representation and classification that utilizes n-grams and a continuous bag of words to address out-of-vocabulary words and capture morphological information. We applied supervised FastText, training on word embeddings for 100 epochs.

mBERT: The original study found mBERT (bert-base-multilingual-uncased) to be the top-performing model on their dataset. mBERT

shares the same architecture as the original BERT, but is trained on a multilingual corpus from Wikipedia encompassing 104 languages.

mDeBERTa-v3: Building upon BERT, the original DeBERTa introduced disentangled attention, where each token is associated with separate embedding vectors for its position and content. DeBERTa-v3 [13] supersedes the masked language modeling task with replaced token detection, akin to ELECTRA's training. Benchmarks indicate that DeBERTa-v3, including its multilingual variant mDeBERTa-v3, stands as one of the most robust models among BERT-like models.

XLM-RoBERTa: We also tested whether larger BERT-like models significantly impact performance. Our resources allowed for the use of XLM-RoBERTa Large [11], a model with 3.5 billion parameters. To understand the impact of model size on performance, we included both the large and the base versions of XLM-RoBERTa in our evaluation.

Fine-tuning: Instead of fine-tuning all parameters, we employed adapters, specifically bottleneck adapters [21] placed after the fully connected layer in each transformer block, from the *Adapters* library. These adapters allow for significant parameter reduction while maintaining performance close to full fine-tuning.

We sourced pre-trained models from HuggingFace and fine-tuned them using PyTorch. Hyperparameter sweeps were conducted across different learning rates to ensure fairness in our comparisons. The chosen hyperparameters were as follows:

Table 3: Hyperparameter settings for model fine-tuning

Hyperparameter	Value
Optimizer	AdamW
	Weight Decay: 0.1, $\beta_1 = 0.9$, $\beta_2 = 0.999$
Batch Size	16
Learning Rate	RoBERTa Large: $1e-4$
	mDeBERTa-v3: $3e-4$
	mBERT, RoBERTa Base: $2e-4$
Adapter Type	Bottleneck adapter per layer
Dropout	0.1

4.2 Hate speech detection

Baselines: We employ the same baselines as in the toxic speech detection task.

Model: We selected XLM-RoBERTa Large, the top-performing model from the toxic speech detection task, and fine-tuned it with adapters for hate speech detection. Given the similarities between the tasks, a model effective in one is likely to excel in the other.

Fine-tuning the hyperparameters for hate speech detection led to the following adjustments:

- *learning rate:* 0.00027
- *train set balance*: 0.719

The train set balance represents the proportion of undersampled non-hateful comments to achieve an optimal balance between classes. Starting with a 1:1 ratio, we reintroduced 71.9% of the undersampled comments to attain the best results, ensuring a robust representation of both classes that mirrors the validation set distribution more closely than a perfect 1:1 balance, which is the original paper's methodology.

4.3 Hate Speech Target Group Detection

Target group detection is a multi-label classification task. This classification problem aims to detect one or more target groups of the hate speech as shown in Figure 5. This task was not done before on this dataset. However, there are similar tasks in the literature such as news category detection [6] and classification of drug trafficking based on Instagram hashtags [15]. Since this task is multi-label, we train a classifier per class in many cases and compare one class vs all remaining. Additionally, we train only on the comments containing hate speech.

FastText: We again use FastText as a baseline, trained for 100 epochs.

BERT: Bidirectional Encoder Representations from Transformers (BERT) is an encoder of the transformer that uses self-attention [9], see Figure 7. There are two architectures, BERT base and large with 12 and 24 encoder layers respectively. Initially, the idea was to tackle this problem with a BERT model since it is the current state-of-the-art approach - a BERT-style transformer architecture with a linear classifier layer on top. We have tried different BERT base models that are pre-trained for hate speech detection for German [1, 18]. We freeze all layers except the last and train the model 10 epochs for the task. We have tried both predictions of one class at a time and a multi-label classification, but BERT needs a large amount of data. Unfortunately, some of the target classes are underrepresented in the dataset and this leads to unsatisfactory results, see Section 5.

Zero-shot learning: Zero-shot learning is a task where a model is trained on labeled data and is then able to classify examples from unseen classes. We tried the task-aware representation of sentences (TARS) [12] that open-source on Flair [10]. It uses cross-attention between text and label. The input sequence consists of the class label and the text to classify that passed through all self-attention layers in BERT.

Few-shot learning: Few-shot learning is the task where a model makes predictions after seeing a limited number of samples. There is a lack of data for some text classification tasks since it need manual labeling. In such cases, an approach is to transfer knowledge from an existing model for one classification task to initialize the weights for a model for the new classification task. We use TARS [12], similar to the zero-shot, and train it for each class for 30 epochs. We tried both multi-class (one class per comment) and multi-label (multiple classes per comment) settings. Multi-class was tried because most of the comments have one or two target classes. The idea was that the model might generalize better.

LLMs: Large Language Models (LLMs) are used for text generation. They consist of large pre-trained transformer models trained to predict the next word (token) given some input text [17]. It uses autoregressive generation to generate text by iteratively calling a model with its own generated outputs, given a few initial inputs. The Llama 2 [23] is a family of pre-trained and fine-tuned LLMs, ranging in scale from 7b to 70b parameters (7b, 13b, 70b). We tried Llama 2 7b [23] with LoRA (Low Rank Adaptation) [16] for the

target class prediction with and without Chain of Thoughts (CoT). The original comments and labels were prepossessed, and a prompt was generated:

"INSTRUCTION: Hate speech is any kind of offensive or denigrating speech against humans based on their identity. Hate speech can be targeted towards gender, age, sexuality, religion, nationality, disability, social status, political views, appearance, or other characteristic. INPUT: What is 1 or more targets of this comment COMMENT*? THINK STEP BY STEP (COT) but use only the following targets: gender, age, sexuality, religion, nationality, disability, social status, political views, appearance, and other. OUTPUT:* CLASSES*".*

The CoT prompt includes the "Think step by step" to encourage the model to reason its decisions.

We load the model with BITSANDBYTES 4-bit quantization [7, 8], and 0.24% trainable parameters. We train for 5 epochs with a batch size of 4.

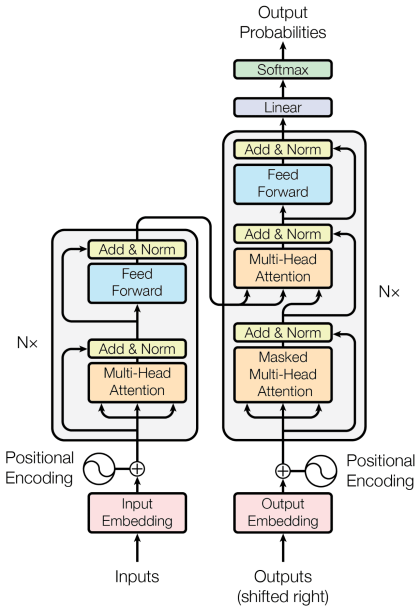

Figure 7: Transformer model

5 Results

In this section, we describe the results for each task.

5.1 Setup

All experiments were conducted on the ETH Euler cluster with the Slurm workload manager. For *toxicity* and *hate speech detection* tasks, a single GPU with either 24GB or 32GB memory was utilized, typically an NVIDIA GeForce RTX 3090 or NVIDIA Tesla V100. The *target group detection* task required one 32GB GPU, except for the LLM experiments, which needed four NVIDIA Tesla V100 GPUs.

The software environment included Python 3.11.2, Pytorch 2.1.1, CUDA 12.1, and Transformers 4.36.

5.2 Toxic speech detection

Here we detail our models' performance on the toxic speech detection task, discussing both primary metrics and additional analyses.

Metric choice: We prioritized area under the PR curve (AUPRC) and F1 score[2] due to their relevance for our problem's requirements. AUPRC was chosen for its combined emphasis on precision and recall for positive examples and its independence from any specific decision threshold. The F1 score is reported at a validation set-optimized threshold; we report it mainly to facilitate comparison with human baselines.

Baselines: Our baselines included FastText and a *Constant* predictor, as previously described. We also used datasets labeled by multiple humans for an indirect comparison, detailed in Section 3.3.

For the CODERS dataset, we evaluated each coder's F1 score against a majority vote-based proxy ground truth. Figure 8 presents these scores along with an aggregate value. This analysis serves as a benchmark for non-expert performance in this task.

For the EXPERTS dataset, we considered the consensus label as ground truth and compared each expert's performance against it. Figure 8 also depicts the experts' F1 scores, showing a significant gap between experts and non-experts.

Our best model's performance on the MAIN set is included in the same figure. It is important to note that the non-experts, experts, and our model were each assessed on distinct datasets, so the comparisons only offer an indirect insight into their relative performance.

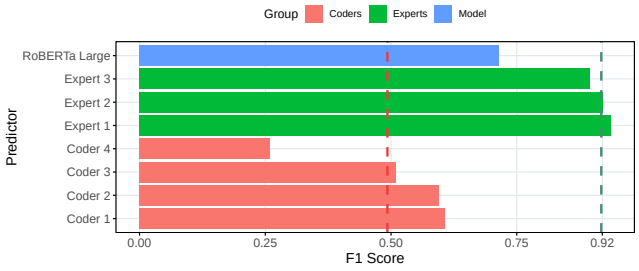

Figure 8: F1 scores for experts, coders, and our best model (evaluated on different datasets, mean scores for groups are dashed)

Results: Table 4 lists the performance outcomes for our models alongside the baselines. XLM-RoBERTa Large consistently surpassed the other models in performance.[3] Given that XLM-RoBERTa Base did not exhibit similar superiority, we attribute the success to model size rather than the architecture itself. Among the smaller models, mDeBERTa-v3 was the top performer, closely rivaling the larger RoBERTa. Thus, mDeBERTa-v3 would be the recommendation for the challenge givers if inference time is a constraint.

[2]F1 score refers to the positive class F1 score. Averaged F1 scores are specified explicitly, including the averaging method.

[3]The F1 scores on the EXPERT set appear lower when using a decision threshold optimized for the validation set, due to their differing label distributions. However, when we recalibrate the threshold specifically for the EXPERT set, the F1 scores improve significantly. The adjusted F1 score for XLM-RoBERTa Large on the EXPERT set is shown in parentheses.

The F1 score on the MAIN set is on par with the human baselines, positioning the model between non-expert and expert performance. This is a significant achievement considering the training data was annotated by non-experts.

Differences in performance: The performance disparities across evaluation sets are evident in the table. To understand the model's behavior, Figure 9 depicts the precision-recall trade-off for XLM-RoBERTa Large on each dataset.

The MAIN set exhibits a smooth trade-off between precision and recall, which we attribute mostly to task-specific characteristics.

In contrast, the REPRESENTATIVE set shows a rapid decline in recall with only a modest increase in precision, indicating the model's limited generalizability. This could stem from a different label distribution and potential concept drift due to changes over time in the comments and labeling process. Further investigation is needed to understand these generalization issues fully.

On the EXPERT set, the model demonstrates high precision with a steep initial rise, but recall decreases almost as quickly. This pattern suggests that the model has internalized a narrower definition of toxic speech from its training data, contrasting with the experts' broader interpretation. One reason for this broader definition could be the consensus-based construction of the set; when labels are agreed upon by three experts, the process might sometimes resemble taking the union of their opinions, potentially leading to more comments being classified as toxic. The EXPERT set's unexpectedly high prevalence of toxic comments (see Figure 3) might be a result of this consensus approach. Further research is required to confirm these hypotheses and to understand the precise factors influencing the model's performance on this dataset.

Calibration measures, such as the Expected Calibration Error (ECE), show the model is well-calibrated on MAIN and REPRESENTATIVE (ECE around 0.03), but less so on EXPERT (ECE of 0.25), likely due to the significant label distribution differences highlighted in Figure 3.

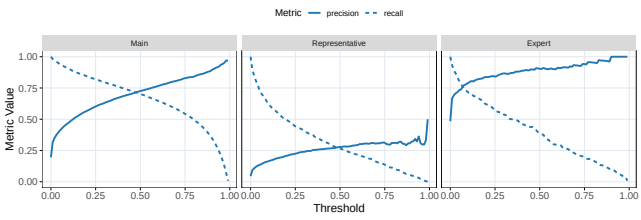

Figure 9: Precision and Recall v. Threshold on different datasets

Inference on a CPU: Considering the potential deployment on CPUs or low-end GPUs, we evaluated the inference performance of our best model, XLM-RoBERTa Large. On an M1 Pro CPU, the model processes approximately 35 comments per second with the Metal backend and 5 comments per second without it. These results suggest that the models are capable of operating within reasonable time frames, particularly on MPS-enabled devices.

5.3 Hate speech detection

This subsection presents our findings for the binary classification of hate speech.

Metric choice: We continue to use AUPRC and F1 Score for the positive class, and additionally report the weighted F1 score, which, despite its atypical use in binary classification, is necessary for comparison with the original paper.

Baselines and results: We employ the same baselines as before: FastText and a *Constant* predictor. Our experiments focus on XLM-RoBERTa Large, the top-performing architecture from the toxic speech detection task. Table 5 shows the results, with the weighted F1 score included in parentheses for the MAIN set.

The patterns observed here mirror those from the previous task, with our model surpassing the baselines, provided the decision threshold is adjusted on EXPERT due to significant miscalibration. The performance discrepancies across the three datasets are also notable here. For an in-depth analysis of these differences, we direct readers to the prior discussion, as the same considerations apply.

We do note, however, that the performance on all the evaluation sets is worse compared to the previous task. We believe this is caused by hate speech being harder to detect than toxic speech; the model probably is probably struggling to learn the differences between these two closely related categories, though this should be investigated more in future research.

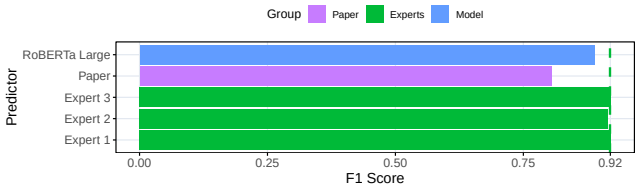

Figure 10: Weighted F1 scores for experts, our best model, and the original paper (evaluated on different datasets, group means dashed)

While we only compute an indirect human baseline from the EXPERT set, Figure 10 compares it to our model and the findings from the original paper. Despite each entity being evaluated on distinct datasets, we posit the comparison is fair as each dataset represents a realistic performance domain for the respective predictor.

5.4 Hate Speech Target Group Detection

In this subsection, we report the performance of the hate speech target group detection task. Additionally, we discuss the time and computational resource requirements for the implemented methods.

Metric choice: We consider two metrics macro and binary F1 and report them for each class. Binary F1 is a score for the positive class only while macro F1 is an unweighted average for each label. The macro F1 is chosen because it does not take label imbalance into account. The binary F1 score represents the actual ability of the model to predict particular target classes.

Table 4: Toxic speech detection results

Model	Threshold for F1	MAIN AUPRC	MAIN F1	REPR. AUPRC	REPR. F1	EXPERT AUPRC	EXPERT F1
baseline: Constant	—	0.195	0.326	0.044	0.085	0.484	0.652
baseline: FastText	0.5	0.295	0.523	0.058	0.086	0.533	0.153
mBERT	0.35	0.751	0.688	0.171	0.236	0.805	0.504
mDeBERTa-v3	0.4	0.766	0.701	0.184	0.254	0.824	0.544
XLM-RoBERTa Base	0.4	0.734	0.677	0.1501	0.192	0.7744	0.414
XLM-RoBERTa Large	0.45	**0.779**	**0.714**	**0.205**	**0.281**	**0.846**	0.588 (**0.777**)

Table 5: Hate speech detection results

Model	MAIN AUPRC	MAIN F1 (weighted)	REPR. AUPRC	REPR. F1	EXPERT AUPRC	EXPERT F1 (optim. thr.)
Baseline: Constant	0.249	0.249	0.041	0.079	0.288	0.447
Baseline: FastText	0.254	0.393 (0.836)	0.056	0.070	0.384	0.139
XLM-RoBERTa Large	**0.650**	**0.612 (0.8891)**	**0.2083**	**0.2699**	**0.604**	0.444 (**0.623**)

Results: We use supervised FastText as a baseline. Additionally, we include the results of dehateBERT [18] (a German BERT fine-tuned for hate speech), TARS few-shot classifier [12], and Llama 2 7b [23] with and without CoT. Table 6 and Figure 11 show the resulting performance of all models that were trained. We can see that TARS few-shot learning and Llama 2 7b outperform other methods. Both models perform similarly for nationality, religion, gender, and political views. These categories are learned and the achieved macro F1 is above 0.8. However, for some categories, the performance is far from satisfactory, e.g., sexuality, age, social status, and disability. We assumed that class imbalance (more negative examples than positive) could be the source of bad performance. To check this assumption, we experimented with the TARS few-shot learning model by balancing the data by a minority class and changing the task to a multi-label classification as shown in Table 8. Both balancing the data (by minority class) and multi-class setting lead to worse performance. We also tried to visualize trained FastText word embeddings. There is no difference between different classes and no clear clusters for each class as expected. Therefore, we believe that class imbalance is not the main problem. There is not enough data for some categories and the existing comments/labels are too heterogeneous.

To experiment with different prompts, we fine-tuned Llama 2 7b with prompts containing CoT and without. Table 7 shows that CoT leads to better results for the underrepresented categories. However, without CoT better-performing categories insignificantly drop in performance.

Runtime: Another important metric to consider is training, inference time, and resource requirements. Table 9 shows the time it takes to fine-tune and perform inference for the different models, and the number of GPUs utilized. We can see that TARS few-shot learning needs significantly more time since it is trained for each category in comparison to Llama or dehateBERT. However, even the smallest Llama 2 7b requires 4 expensive GPUs while the TARS few-shot needs only one. Additionally, the Llama inference takes

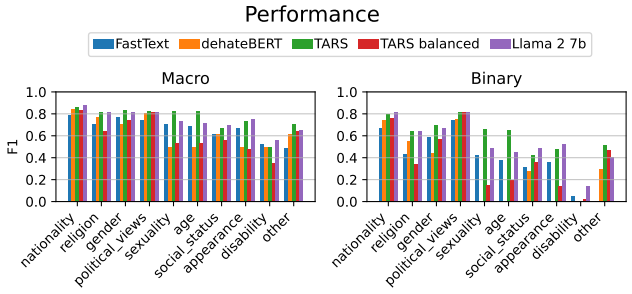

Figure 11: Target group detection performance

approximately 6 hours while the TARS few-shot needs a few minutes. It is important to mention, that we have tried to experiment with the bigger Llama 2 70b but because of memory constraints and Euler cluster limitations, it was not possible. Therefore, with limited resources, the TARS few-shot is the optimal solution.

6 Discussion

The exploration of model performance across different datasets has led to several insightful observations. Perhaps the most notable is the pronounced disparity between the MAIN and REPRESENTATIVE sets. This difference can be largely attributed to the data collection methodology. For the MAIN set, where the data was actively filtered using machine learning models before labeling, our models achieved satisfactory performance.

Our models tend to outperform trained research assistants in tasks with available human baselines while falling short of expert-level performance. This outcome aligns with expectations considering that the research assistants, and not the experts, performed the labeling of our training data. It is encouraging to see that our models exceed the performance of the original paper's model, providing the challenge givers with a tangible improvement for their existing tasks. Our contribution is particularly significant for the target group detection

Table 6: Results for target class detection

	FastText		dehateBERT		TARS few-shot		Llama 2 7b	
Model	Macro	Binary	Macro	Binary	Macro	Binary	Macro	Binary
Nationality	0.785	0.668	0.841	0.741	0.864	0.795	**0.877**	**0.815**
Religion	0.711	0.437	0.769	0.555	**0.815**	**0.642**	0.813	0.641
Gender	0.775	0.584	0.703	0.438	**0.834**	**0.699**	0.813	0.666
Political views	0.747	0.739	0.808	0.752	**0.821**	**0.820**	0.813	0.818
Sexuality	0.706	0.422	0.496	0.0	**0.827**	**0.660**	0.738	0.491
Age	0.684	0.382	0.496	0.0	**0.823**	**0.655**	0.714	0.449
Social status	0.615	0.315	0.611	0.275	0.672	0.422	**0.701**	**0.486**
Appearance	0.673	0.358	0.495	0.0	0.736	0.483	**0.755**	**0.525**
Disability	0.524	0.054	0.498	0.0	0.497	0.000	**0.564**	**0.137**
Other	0.485	0.0	0.613	0.299	**0.704**	**0.518**	0.653	0.406

Table 7: Different Llama 7b prompts

	Simple prompt		CoT prompt	
Model	Macro	Binary	Macro	Binary
Nationality	**0.877**	**0.815**	0.869	0.806
Religion	**0.813**	**0.641**	0.741	0.506
Gender	**0.813**	**0.666**	0.789	0.627
Political views	0.813	0.818	**0.834**	**0.835**
Sexuality	**0.738**	**0.491**	0.682	0.388
Age	**0.714**	**0.449**	0.688	0.406
Social status	0.701	0.486	**0.723**	**0.514**
Appearance	0.755	0.525	**0.789**	**0.590**
Disability	**0.564**	0.137	0.563	**0.145**
Other	0.653	0.406	**0.705**	**0.504**

Table 8: Results of diverse approaches for TARS few-shot

	Multi-label		Balanced		Multi-class
Model	Macro	Binary	Macro	Binary	Binary
Nationality	**0.864**	**0.795**	0.834	0.765	0.270
Religion	**0.815**	**0.642**	0.639	0.342	0.028
Gender	**0.834**	**0.699**	0.747	0.567	0.102
Political views	**0.821**	**0.820**	0.819	**0.820**	0.572
Sexuality	**0.827**	**0.660**	0.529	0.147	0.026
Age	**0.823**	**0.655**	0.529	0.195	0.040
Social status	**0.672**	**0.422**	0.563	0.363	0.131
Appearance	**0.736**	**0.483**	0.481	0.145	0.039
Disability	**0.497**	0.000	0.351	**0.025**	0.000
Other	**0.704**	**0.518**	0.645	0.467	0.000

task as we offer a new model for a task that previously lacked automated solutions.

However, the results should be seen in the context of the limitations imposed by various factors. One such factor is the nature of the dataset itself, which, being labeled by non-experts, might not capture the full complexity and nuance of hate speech and toxic comments. Furthermore, the discrepancies in label distributions between training and evaluation sets suggest that model generalization ability remains a challenge.

Table 9: Runtime of different methods

Model	Time [m]	Comp. units
FastText	5.2	CPUs
dehateBERT	829	1 RTX 3090 24GB
TARS few-shot	2126	1 RTX 3090 24GB
Llama 2 7b	550	4 V100 32GB

In the discussion of our results (Subsection 5.4), we highlighted the data scarcity for certain classes within the target class detection task. This imbalance presents a significant challenge, as underrepresented classes lack the volume of examples necessary for the models to learn effectively. Despite our various experiments to mitigate this issue, performance gains were limited. Consequently, acquiring more data for these sparse classes is likely to be beneficial. On a positive note, the classes that do possess ample data exhibit satisfactory performance, underscoring the importance of a well-populated dataset for robust model training.

Computational resources also represent a constraint in our study. We ensured that the most effective methods identified for each task were compatible with the processing capabilities provided by the challenge givers. This consideration allowed us to develop solutions that are feasible within the given CPU and GPU limits. Furthermore, we experimented with more resource-intensive models to understand the relationship between computational demand and model performance. These insights inform the practical application of our findings and highlight the importance of optimizing for computational efficiency.

6.1 Applications and Future Work

Currently, the models are set to be utilized in two main ways. The first is the immediate deployment by our challenge giver partners to monitor hate speech in online comments. The second is to aid ongoing research aimed at understanding and curbing online hate speech. One notable project is the development of a real-time dashboard, which will track and display the dynamics of online discourse,

quantify hate speech, and identify targeted groups, as well as how these elements fluctuate with global events.

For both applications, ease of deployment and the capacity for ongoing fine-tuning are crucial to keep the models up-to-date and mitigate concept drift. By hosting our models on HuggingFace, we've streamlined the process for continuous improvement with new data.

Looking forward, enhancing the models could involve collecting additional data for underrepresented classes. Incorporating uncertainty prediction may also be valuable, potentially aiding in practical deployment decisions. Furthermore, further experimenting with larger models like Llama or other large language models could yield performance gains, particularly if prompts are well-tuned; as we have shown within this report, prompt-tuning can greatly impact model performance.

Analytical methods such as text mining and chi-squared analysis could provide insights into the lexicon associated with different hate speech target groups. This information might reveal why certain classes are more readily detectable, such as those with distinctive keywords. Similarly, analyzing the comments that challenge the models the most could uncover common characteristics, informing future research and model refinement.

7 Conclusions

This report outlined the Hate Speech Detection Project, tackling toxic speech detection, hate speech classification, and identification of targeted groups within hate speech. We introduced a robust data preprocessing strategy, assessed the dataset's composition, and addressed the challenges associated with its collection and quality.

Our fine-tuned models demonstrate superior performance compared to the state-of-the-art and are designed to operate within resource-limited environments. Introducing a target class detection classifier represents a novel contribution and shows promising performance for well-represented categories. While the performance of minority classes could benefit from additional data, the current models serve as a strong foundation for future enhancements and practical applications.

8 Acknowledgments

We thank our coach Dr. Alexander Ilic, and challenge givers Dr. Philip Grech, and Dr. Nicolai Berk from the Immigration Policy Lab for the time invested in our meetings and helpful advice. We also thank Dominik Stammbach for mentoring and suggestions.

References

[1] Sai Saketh Aluru, Binny Mathew, Punyajoy Saha, and Animesh Mukherjee. 2020. Deep Learning Models for Multilingual Hate Speech Detection. arXiv:2004.06465 [cs.SI]

[2] Lukas Biewald. 2020. Experiment Tracking with Weights and Biases. https://www.wandb.com/ Software available from wandb.com.

[3] Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching Word Vectors with Subword Information. arXiv:1607.04606 [cs.CL]

[4] Berkman Klein Center. 2016. Harmful Speech. https://cyber.harvard.edu/node/99714 [Online].

[5] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer. 2002. SMOTE: Synthetic Minority Over-sampling Technique. *Journal of Artificial Intelligence Research* 16 (June 2002), 321–357. https://doi.org/10.1613/jair.953

[6] Shahzada Daud, Muti Ullah, Amjad Rehman, Tanzila Saba, Robertas Damaševičius, and Abdul Sattar. 2023. Topic Classification of Online News Articles Using Optimized Machine Learning Models. *Computers* 12, 1 (2023). https://doi.org/10.3390/computers12010016

[7] Tim Dettmers, Mike Lewis, Younes Belkada, and Luke Zettlemoyer. 2022. LLM.int8(): 8-bit Matrix Multiplication for Transformers at Scale. *arXiv preprint arXiv:2208.07339* (2022).

[8] Tim Dettmers, Mike Lewis, Sam Shleifer, and Luke Zettlemoyer. 2022. 8-bit Optimizers via Block-wise Quantization. *9th International Conference on Learning Representations, ICLR* (2022).

[9] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *CoRR* abs/1810.04805 (2018). arXiv:1810.04805 http://arxiv.org/abs/1810.04805

[10] FLAIR. [n. d.]. FLAIR. https://github.com/flairNLP/flair [Online].

[11] Naman Goyal, Jingfei Du, Myle Ott, Giri Anantharaman, and Alexis Conneau. 2021. Larger-Scale Transformers for Multilingual Masked Language Modeling. arXiv:2105.00572 [cs.CL]

[12] Kishaloy Halder, Alan Akbik, Josip Krapac, and Roland Vollgraf. 2020. Task-Aware Representation of Sentences for Generic Text Classification. In *Proceedings of the 28th International Conference on Computational Linguistics*, Donia Scott, Nuria Bel, and Chengqing Zong (Eds.). International Committee on Computational Linguistics, Barcelona, Spain (Online), 3202–3213. https://doi.org/10.18653/v1/2020.coling-main.285

[13] Pengcheng He, Jianfeng Gao, and Weizhu Chen. 2023. DeBERTaV3: Improving DeBERTa using ELECTRA-Style Pre-Training with Gradient-Disentangled Embedding Sharing. arXiv:2111.09543 [cs.CL]

[14] Sophie Henning, William Beluch, Alexander Fraser, and Annemarie Friedrich. 2023. A Survey of Methods for Addressing Class Imbalance in Deep-Learning Based Natural Language Processing. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, Andreas Vlachos and Isabelle Augenstein (Eds.). Association for Computational Linguistics, Dubrovnik, Croatia, 523–540. https://doi.org/10.18653/v1/2023.eacl-main.38

[15] Chuanbo Hu, Bin Liu, Yanfang Ye, and Xin Li. 2023. Fine-grained classification of drug trafficking based on Instagram hashtags. *Decision Support Systems* 165 (2023), 113896. https://doi.org/10.1016/j.dss.2022.113896

[16] Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. LoRA: Low-Rank Adaptation of Large Language Models. arXiv:2106.09685 [cs.CL]

[17] HuggingFace. [n. d.]. LLM Tutorial. https://huggingface.co/docs/transformers/llm_tutorial [Online].

[18] HuggingFace. 2022. Hate-speech-CNERG/dehatebert-mono-german. https://huggingface.co/Hate-speech-CNERG/dehatebert-mono-german [Online].

[19] Ana Kotarcic, Dominik Hangartner, Fabrizio Gilardi, Selina Kurer, and Karsten Donnay. 2023. Human-in-the-Loop Hate Speech Classification in a Multilingual Context. arXiv:2212.02108 [cs.CL]

[20] United Nations. 2019. What is hate speech? https://www.un.org/en/hate-speech/understanding-hate-speech/what-is-hate-speech#:~:text=To%20provide%20a%20unified%20framework,person%20or%20a%20group%20on [Online].

[21] Jonas Pfeiffer, Ivan Vulić, Iryna Gurevych, and Sebastian Ruder. 2020. MAD-X: An Adapter-Based Framework for Multi-Task Cross-Lingual Transfer. arXiv:2005.00052 [cs.CL]

[22] David Sayce. 2016. dsayce.com. https://www.dsayce.com/social-media/tweets-day/#:~:text=Every%20second%2C%20on%20average%2C%20around%206%2C000%20tweets%20are%20tweeted%20on,200%20billion%20tweets%20per%20year. [Online].

[23] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. LLaMA: Open and Efficient Foundation Language Models. arXiv:2302.13971 [cs.CL]

[24] Evzen Wybitul. 2023. Data Processing Pipeline for Hate Speech Recognition. https://github.com/Eugleo/hate-speech-data-preprocessing [Online].