Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Hate Speech Classifier

Data Science Lab Project Report

Rahul Steiger (rasteiger), Shakir Yousefi (syousefi), Flavia Pedrocchi (flaviap)

December 20, 2023

Alex Ilic (ETH AI Center), Philip Grech (IPL)

Department of Computer Science, ETH Zürich

Abstract

The proliferation of online hate speech has become a growing issue that needs to be addressed [2]. Its detection is especially relevant to online platforms to facilitate moderation and prevent the spread of hateful messages. We investigate the detection of hate speech and its target group with different architectures and training. Our best model for both detecting hate speech and pinpointing the target group was the fully fine-tuned E5 model. However, the LoRA-based approach cut training time in half while maintaining 95% efficacy. Finally, we believe that methods we investigated such as Integrated Gradients or the automatic removal of toxicity with Llama could be effective approaches to actively confront and reduce the impact of hate speech

Contents

Chapter 1

Introduction

While hate speech is hard to define consistently, most definitions agree that it disparages or discriminates against individuals or groups based on attributes such as ethnicity, religion, gender, or others. This speech often incites hatred, hostility, or violence towards those targeted groups or individuals. Thus, recognizing hate speech can help mitigate its effect, and although most online platforms already have moderation staff, machine learning models can help identify and remove hateful comments at a larger scale.

This work looks at different architectures and approaches for hate speech prediction and target identification. We experimented with different features to improve training cost or performance such as the implementation of LoRA and the use of novel optimizers. Importantly, we also investigate different possibilities to counter hate speech online. In general, this work aimed to explore many options and push performance while maintaining practical relevance. Although there have been many successful approaches to hate speech classification, few of them focus on Switzerland's social and multilingual context and aim to be deployed and trained at a reasonable cost. A relevant paper by A. Kotarcic et al. [7] makes use of the same dataset yet we do not fully agree with their data set balancing and choice of evaluation metric. Also, their work does not incorporate such an extensive analysis of different classification methods.

Chapter 2

Background

The rise of the internet and social media has changed the world. It dictates how we interact with each other, access information from the news or organize to demand political change. Importantly, it has also established a new means for hate speech to spread. Regulating what information or ideas have a place on the web has become a central topic of debate [13]. The particular challenges in regulating online hate speech stem from its unique features such as anonymity, rapid dissemination, enduring nature, and cross-jurisdictional reach [5].

Ultimately, the task of moderation falls on the social media or newspaper companies and most online platforms do outlaw hate speech in their terms of service. Notably, in 2016 Facebook, Microsoft, Twitter and YouTube agreed to a European Union Code of Conduct to prevent and counter the spread of illegal hate speech online by removing or disabling access to such content in less than 24 hours [17]. Methods utilized by these companies to tackle hate speech involve user reporting, Artificial Intelligence flagging, and the manual review of content by staff [5]. Due to the sustained growth of internet content and user number, automated moderation by AI will become more and more relevant.

However, it is challenging to find a consistent definition for hate speech and its subjective nature as well as context nuances make it hard to identify. Especially Switzerland has a very complex multilingual and sociopolitical context that can be hard to grasp by AI. Therefore, the need arises for a model tailored to Switzerland's environment that can identify and label hate speech in online comments.

An architecture that is capable of detecting hate speech and its targets could have several applications. It could help filter hateful comments online but also give feedback such as highlighting the problematic part of the sentence or automatically proposing a more civil alternative to the user. Understanding the context of the comment such as the target group could also make counter-speech more effective.

Chapter 3

Methods

3.1 Definition of Hate Speech

The definition of hate speech can vary across different jurisdictions, cultures, and contexts, leading to some inconsistency in how it's interpreted and addressed. Generally, hate speech involves any kind of communication that attacks or uses pejorative or discriminatory language with reference to a person or a group on the basis of who they are, such as on their religion, ethnicity, nationality, race, colour, descent or gender [12]. While hate speech is a subset of toxic speech that targets specific groups based on certain characteristics, toxic speech covers a broader spectrum of harmful communication that includes various forms of offensive or damaging language and behaviour beyond just targeting specific groups.

Figure 3.1: Hate speech is a type of toxic language

3.2 Swiss Hate Speech Corpus

The Swiss Hate Speech Corpus is composed of 422'046 labelled comments taken from online newspaper comments from different media outlets and tweets collected using the Twitter API of "politically interested users", meaning accounts following at least five Swiss newspapers or politicians. The newspaper comments include not only published but also moderated and deleted comments. Though the majority of comments are in German, there is also a significant number in French, which calls for a multilingual model.

In this work we have labels for both targeted (hate speech) and untargeted toxic speech. Although the primary focus remains on the hate speech classification of comments, the additional information given by the toxicity label can still be useful for training the model. Further, the dataset includes the specific target group for any comments that consist of hate speech. It is relevant to mention that many different annotators were involved in the labelling process and due to the subjective nature of hate speech this can lead to inconsistencies in data that affect the model performance.

We also acquired a dataset of examples annotated by experts (faculty & professors) who have a more precise understanding of the hatespeech definition and so lessen this variability.

3.2.1 Target Group Distributions

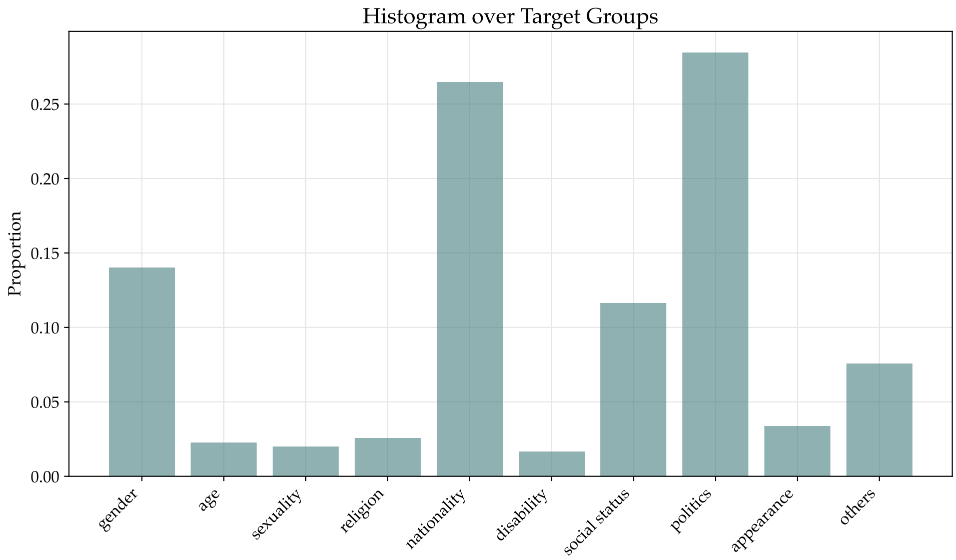

Figure 3.2: A histogram over the different target groups in the Swiss Hate Speech corpus.

As is seen on Figure 3.2, the groups 'politics' and 'nationality' have the highest proportion of comments. More importantly, we see that the remaining target groups are relatively sparse compared to these two. This naturally aligns with the fact that certain target groups are affected differently in regards to hate-speech.

Furthermore, it is important to note that a comment can belong to multiple target groups. On Figure 3.3, we see a histogram over the distribution. It is noteworthy that most of the hate-speech comments only have one target group, and the number of target groups decreases drastically.

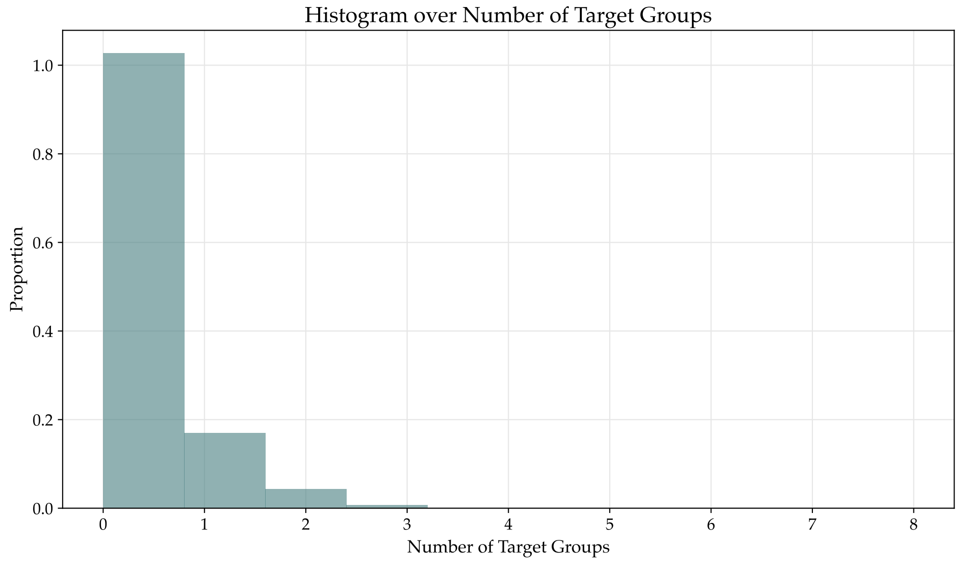

Figure 3.3: A histogram over the aggregated count of target groups in the Swiss Hate Speech corpus.

Considering this, it is natural to assume that the feasability of identifying the target groups is not the same across different target groups. Specifically, we could already expect that a model would perform far better on '`politics`' and '`nationality`' than the remaining target groups.

3.2.2 Preprocessing

Data Cleaning

We performed some routine data cleaning on the corpus by dropping all the NaN values and duplicate entries. Then, we removed all potential overlaps between our standard and expert datasets and labelled the comments with their corresponding language (French/German).

Text Processing

The Naïve Bayes model needed some additional text processing to perform well. Extra white spaces, line breaks, htmls and @-mentions were removed, emojis transcribed to words and words converted to lowercase as done by A. Kotarcic et al. [7]. Additionally, we removed punctuation, digits and stop words using the re and nltk libraries. Then, we performed lemmatization with the German and French spaCy pipelines de_core_news_md and fr_core_news_md respectively. Finally, the text was tokenized via Term Frequency - Inverse Document Frequency (TF-IDF).

Data Split

Our data split consists of 80% training, 10% validation, and 10% test dataset. Each dataset has the same ratio of hate speech, German, and French comments. In total we create three splits, each with a different random seed.

3.3 Natural Language Inference

An approach we will utilize in both classification of hate-speech, and target group identification is natural language inference (NLI), also called textual entailment (TE). It defines a directional relation between two different text fragments.

More specifically, the two text fragments are denoted a `hypothesis`, H, and a `premise`, P. The task of natural language inference is now to determine whether $P \Rightarrow H$, i.e. given P is true, does this entail H? We emphasize that this is agnostic to whether or not the `premise` is actually true. Furthermore, this differs from pure logical entailment, and the outcome should be interpreted as, if a human reads P, does H follow.

We denote the combination of a hypothesis and a premise, a (`hypothesis`, `premise`)-pair. The assertion, $P \implies H$, is the outcome of the pair and can belong to one of three categories, $\{\texttt{entailment}, \texttt{contradiction}, \texttt{neutral}\}$. To exemplify the three different outcomes, we have the following example in table 3.1. We leave the explanation of the outcome for each (`hypothesis`, `premise`)-pair in Appendix B.1.

`Premise`, P	`Hypothesis`, H	`Outcome`, $P \Rightarrow H$
Reducing carbon emissions is crucial for slowing down global warming.	Lowering the amount of greenhouse gases will help combat climate change.	`entailment`
Regular exercise is beneficial for maintaining good health.	Being physically active has no impact on a person's health.	`contradiction`
Reading books regularly can expand your knowledge and improve your vocabulary.	Reading science fiction novels is the best way to relax.	`neutral`

Table 3.1: Examples of `entailment`, `contradiction`, and `neutral`

3.3.1 Natural Language Inference for Classification

While the task of NLI is not directly related to either task, recent efforts in natural language processing have shown we can incorporate the NLI structure into a classification task. Specifically, Transformer models [20] employ self-attention, which also use positional encodings. This is relevant for the NLI task, because this allows using seperate positional encodings for the hypothesis and premise — effectively enabling the model to distinguish between the two.

In the case of the Swiss Hate Speech corpus, a natural approach for the premise is simply using the comment. For both tasks, we restrict ourselves to only using the outcomes $\{\texttt{entailment}, \texttt{contradiction}\}$. This casts the task of NLI into a binary-classification problem. Thus, as an objective function, we can use binary-cross-entropy loss:

$$J(\theta) = -\frac{1}{N}\sum_{i=1}^{N} y_i \log(\hat{y}_i) + (1 - y_i)\log(\hat{y}_i) \tag{3.1}$$

where N is the size of the dataset, and y_i is a binary variable that is encoded by the NLI outcome. In other words, if an entailment follows, we set $y = 1$ (and $y = 0$ if it is a contradiction).

The specifics on how to generate the (`hypothesis`, `premise`)-pairs are dependent on each task, and we refer to subsequent sections 3.5.2 and 3.6 for more information.

Furthermore, the NLI approach is not specific to any Transformer architecture as long as the input correctly uses specific encodings for the hypothesis and premise. Hence, this allows us to use a pipeline, where we only have to swap the model weights to use a different architecture.

3.4 Hate Speech Classification

To detect whether a comment contains hate speech, we investigate two architectures for our pipeline: Naïve Bayes and Transformer-based models.

3.4.1 Naïve Bayes

Naïve Bayes is often used for text classification due to its speed and simplicity. However, the standard Multinomial Naïve Bayes has severe assumptions on the data that in practice often don't hold. Therefore, we decided to use the Complement Naïve Bayes classifier described by Rennie et al. [15]. Here, systemic errors such as the Skewed Data Bias or Weight Magnitude Errors are corrected, which make it a particularly suited method for imbalanced data sets. In this work we use Naïve Bayes as a baseline to compare to our more computationally expensive transformer-based models. Because this model is not multilingual, separate model were needed to predict German and French comments respectively.

3.4.2 Transformer-based Models

Our standard model consists of a backbone language model with a classification head on top of it. For our backbone model, we consider the pre-trained versions of the popular multi-lingual transformer models BERT [4], RoBERTa [10] and its variation E5 [21]. The classification head is a linear layer. Since the goal of this project was to create a model which could be deployed and trained at a reasonable cost, we did not explore LLMs.

Then, we also explored the use of NLI for hate speech classification. We introduce a hypothesis and premise as described in section 3.3. In this case, our hypothesis is: `"This text contains hate speech"`, our premise is the comment, and the entailment label is whether this column is actually hate speech. Furthermore, this approach also works for the toxicity label by just replacing `hate speech` with `toxicity` in the hypothesis.

Given the substantial computational demands of fully fine-tuning such a model, we investigate the efficacy of a LoRA-based approach [6] in contrast to fine-tuning all parameters. Our choice of a LoRA-based approach is deliberate, as it allows us to capitalize on its efficiency in training only a subset of parameters while still achieving comparable performance to fully fine-tuned models.

Recent work [1] has shown that by choosing a different optimizer from AdamW [11], one could improve the accuracy of vision transformers by 2% on ImageNet. While on par improvements could not be seen with this specific optimizer for language tasks, the Sophia optimizer [9] has shown to lead to significant performance improvements for LLMs. In addition to exploring the effect of LoRA-based approaches, we also seek to determine whether employing a different optimizer can leard to a performance improvement.

3.5 Target Group Identification

After predicting whether a comment contains hate speech, we tackled the task of predicting its corresponding target group. This is a multilabel classification problem, meaning that multiple nonexclusive labels may be assigned to each hate speech comment. Importantly, we isolate this problem from the former prediction by only including comments with a positive hate speech label in this task.

3.5.1 Naïve Bayes

For this task we utilized again the Complement Naïve Bayes classifier as described above. Since this is a multilabel problem, we need one classifier per label to the additional classifier per language resulting in a total of 22 models for prediction. Even though this seems like a huge amount, these models are very simple and efficient to train.

3.5.2 Transformer Model

As mentioned in section 3.3.1, we use an NLI approach using a Transformer model. Here, we generate the (hypothesis, premise)-pair in the following way: We first fix the comment as a premise. Next, the hypothesis is generated by concatenating the string, `"This text targets the user based on "` and the target group. Thus, a single comment corresponds to generating a (hypothesis, premise)-pair for each target group. The label of the corresponding pair is simply whether or not the comment contains the target group. After forming a prediction, we can reshape the individual predictions to obtain a binary vector for each comment, where the corresponding index corresponds to whether or not the target group is included.

3.6 General Purpose Models

In this section we propose a way to combine the previous two tasks, target group identification and hate speech classification, into a single model by utilizing Natural Language Inference.

Combining the NLI data as described in the two previous sections 3.4.2 and 3.5.2 from the hate-speech, toxicity, and target group labels allows us to train a single model which is be capable predicting these three labels.

3.7 Integrated Gradients

We tested the use of Integrated Gradients as described in [16] on our hate speech classifiers. It is a method that explains model predictions by calculating the integral of gradients along a path from a reference point to input features. It helps understand how each feature contributes to the model's prediction and so helps explain their decisions in a more understandable manner. In our case, when a model predicts hate speech this method highlights the words in the sentence that most influenced that decision. Not only does this aid the interpretability of the model, but also opens up the possibility of utilizing this information to give personalized feedback to the user about the most problematic parts of any comment flagged as hate speech.

Chapter 4

Results

4.1 Training setup

We train each model for 3 epochs with 400 warm-up steps. Since we do not use the validation dataset to modify our training process, we use it to together with the training dataset to train the model. We train each model configuration with different seeds on each of the three data splits. For more details about hyper-parameters, refer to the the Appendix C.

4.2 Hate Speech Classification

As one can see in Figure 4.1, using more complex and modern models leads to better results. We would like to emphasize that the RoBERTa and E5 model have the identical model architecture, just different weights.

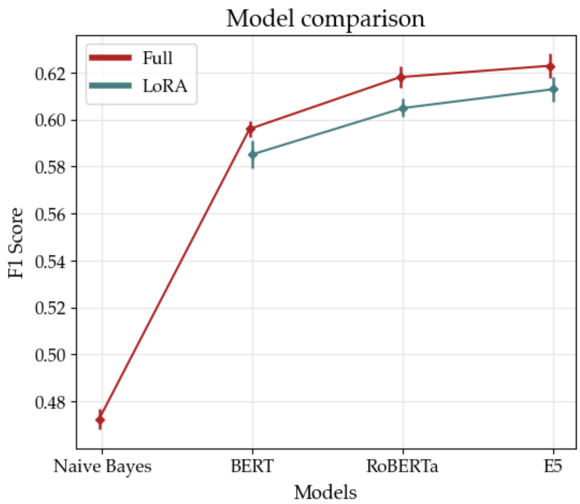

Figure 4.1: F1 Score of Naïve Bayes and the different transformer models with full finetuning or LoRA

In Figure 4.2, one can see the impact of training the E5 model with different datasets from section 3.6. The E5 NLI model was trained only on the toxicity and hate speech task, whereas the E5 NLI dual and E5 NLI dual + were both trained on the toxicity, hate speech, and target group identification task. The difference between these two models is that NLI dual contains only 1 negative example per target group in contrast to the 9 others that NLI dual + contains.

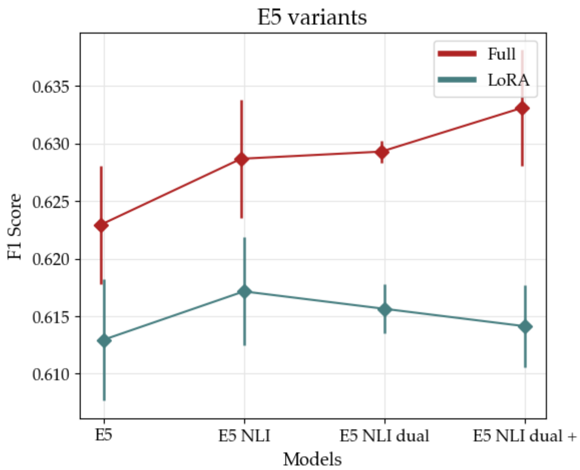

Figure 4.2: F1 score of E5 trained on different datasets with full finetuning or LoRA

We also explore the effect of using the Sophia optimizer instead of Adam. As one can see in Figure 4.3, this leads to a significant performance improvement for full fine-tuning and a decrease in classification performance for LoRA.

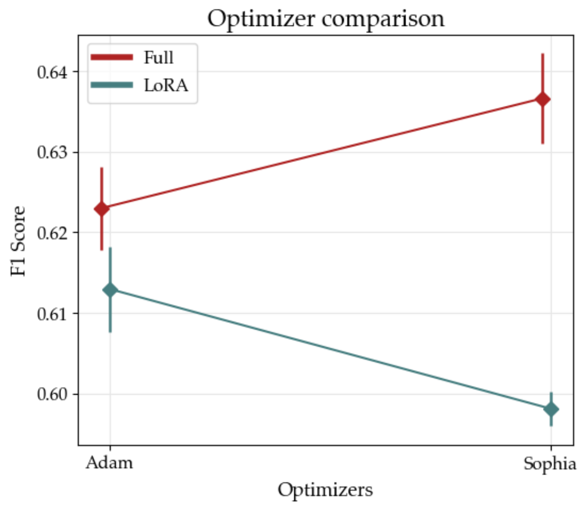

Figure 4.3: F1 score of E5 trained with different optimizers and with full finetuning or LoRA

While Sophia leads to a performance improvement for the pure classifier, the performance improvement for the general purpose models are non-existent non-existent as one can see in Figure 4.4.

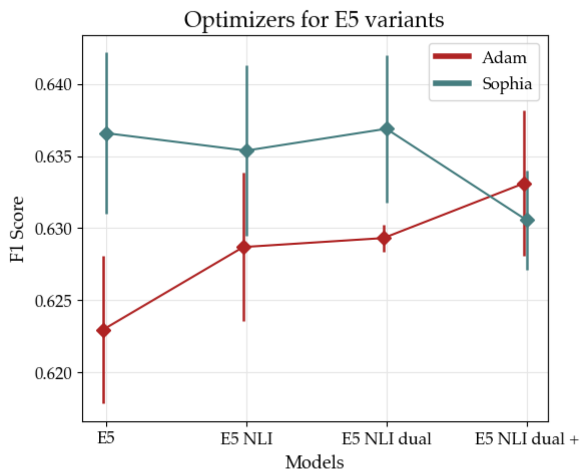

Figure 4.4: F1 score of E5 trained with different optimizers and on different datasets

It is worth mentioning that using a LoRA-based approach for training halved training time and memory requirements while achieving 95% efficacy with regard to performance.

4.3 Target Group Identification

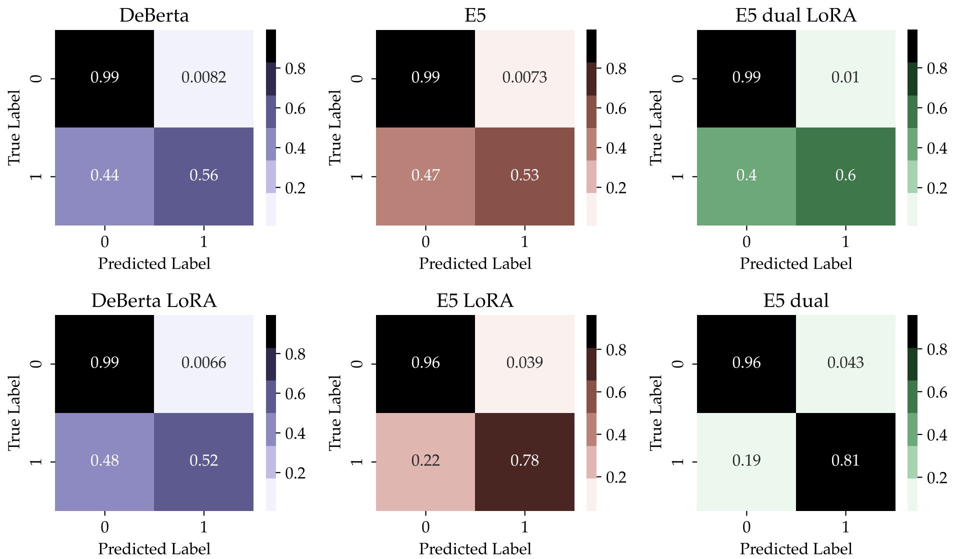

Figure 4.5: Confusion matrix over the different Transformer models for the NLI task.

The performance of the NLI approach on the individual (hypothesis, premise pairs) can be seen on Figure 4.5. We note that the most important factor in target group identification likely is whether the classifier has good performance on label 1. Recall that that we generate pairs for each target group and by Figure 3.3, we see that most of these pairs result in a target label of 0.

On Figure 4.5, it can be seen that E5 performs better than DeBerta. The E5 dual model as described in section 4.2 also shows an improve in performance. The low-rank approximation (LoRA) for the model E5 dual performs better than all other models other than the full-rank version of itself. While the performance of the Transformers seem similar, the small errors in the NLI task propagate to target group identification.

Figure 4.6: Diagram showing how F1-scores vary across target groups

In Figure 4.6, the performance of the confusion matrix is also reflected in the down-stream classifications. By comparing Figure 4.6 with Figure 3.2, we see a strong dependence on the amount of data available for each target group. In particular, the Naïve Bayes model performs almost on par with the Transformer based models for target groups, where there is high amount of data available. Conversely, the Transformer based models perform better on target groups with less data available, and the gap across target groups is smaller. We refer to Appendix B.2 for the full numerical results with three seeds.

4.4 Integrated Gradients

Using Integrated Gradients to highlight the problematic part of a sentence worked quite well when the model was confident about its prediction. This method seems to excel in cases where there was a clear part of the comment that was hateful or when profanities were used. Yet when the comment is more ambiguous or generally offensive it was harder for the method to determine what should be changed and so was of less use. The upside is that as the classification model improves, this method will work more effectively. Thresholding the word importance score could help define a limited number of problematic words.

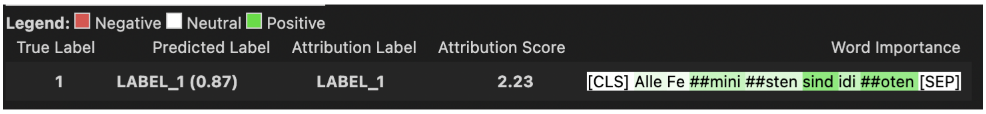

Figure 4.7: Example of the result of using Integrated Gradients on a comment

Chapter 5

Discussion

5.1 Dataset Ambiguity

The definition of toxicity and hate speech is subjective. Especially, the difference between hate speech and toxicity can be confusing and ambiguous in certain cases. This was corroborated by the fact that when a significant majority of the comments our model miss-classified were miss-labeled in our opinion. This was exacerbated by the fact that the dataset was labeled by multiple individuals with their own biases.

We were provided with an expert dataset where each comment was labeled by multiple experts and the labels were determined in a voting fashion. However, this dataset only contains 500 comments and additionally has a completely different distribution regarding the hate speech, toxicity, and target labels. While we report the performance metrics of our models on this dataset, we believe that these are not ideal for comparing models trained with the provided non-expert dataset.

5.2 Performance Discrepancies

While we had significant improvements regarding the hate speech classification performance compared to the Naïve Bayes and Bert baselines, we still do not believe that our models can be used in a production environment. We believe that our low F1 scores are mainly caused by the dataset ambiguity described in section 5.1. Furthermore, we believe that the greatest improvements in hate speech classification performance can be achieved if the ambiguity and number of miss-labeled samples in the dataset is reduced significantly.

The metrics values for the target group identification are a lot higher for the target group identification. However, this is caused by the fact that we only consider the hate speech comments for this task when evaluating the metrics. As one can see in sections 3.2 and 3.3, the metrics for target groups that appear more often in the training data are better. We believe that the greatest improvement in target group identification can be achieved if more training data for underrepresented target groups are provided.

5.3 Target Group Identification

5.3.1 Multi-Label, Multi-Class and Intersectionality

As mentioned in the previous chapters, the comments are not restricted to one target group. Since the number of comments containing multiple target groups are relatively few, one could also reframe it as a multi-class problem in one of few ways: Simply omit comments with more than one label, randomly sample one of the labels/introduce a number of duplicate comments, or introduce some kind of soft-labelling.

Intersectionality

To address the first point, we note that it may be important for the end-user to identify which target groups co-occur. The concept of intersectionality is a framework that identifies multiple factors of an individual's social and political identities. Combining these overlapping identities to understand how systems of oppression and discrimination intersect and influence the experiences and opportunities of marginalized individuals [22]. Therefore, we deem it unreasonable to simply omit the comments with more than one label.

Considerations for Multi-Class in Multi-Label Setting

Using a standard loss function for multi-class classification, like cross-entropy loss, still poses certain challenges for the latter approaches. Here, we denote N the size of the dataset, K the number of classes considered, and y_{ik} a one-hot encoded vector.

$$J(\theta) = -\frac{1}{N} \sum_{i=1}^{N} \sum_{k=1}^{K} y_{ik} \log(\hat{y_{ik}}) \tag{5.1}$$

If we include the same comment with more than one label, we still propagate gradients to the model that penalizes a potentially correct classification. One could argue that by having duplicates in the data set, the effects of this are negligible, but this has to be tested empirically.

Another approach would be to simply include all labels in the one-hot encoded vector, y_{ik}. It is common practice to apply a Softmax operation on the final layer of the neural network:

$$\text{Softmax}(x_i) = \frac{\exp(x_i)}{\sum_j \exp(x_j)} \tag{5.2}$$

This makes the output of the neural network interpretable as probabilities. In this case, the model would try to minimize an objective function, where the optimal solution is not in the feasible set of parameters. This in itself is not a problem, but could be an issue, since we implicitly weigh classifying a comment with one target label differently than one with multiple target labels. Similar issues can arise, if we use soft-labelling.

All of the challenges sketched above can likely be attributed the fact that the task at hand is ill-posed with respect to modelling a categorical distribution. Considering this, we still believe using an NLI approach is reasonable.

5.3.2 Conditioning on Target Groups

When identifying the accompanying target groups, we assumed a priori that the comment contains hate-speech. This is necessarily an easier task than directly inferring the target group, since a target group only exists for hate-speech comments. In light of this, the results obtained for this task should only serve as a hypothetical best-case scenario, and not an indication of real-world performance.

Methodologically, decoupling hate-speech classification and target group identification still makes sense for a list of reasons. It is reasonable to assume that both tasks are not equally feasible to solve. The concept of hate-speech itself can seem more abstract than identifying target groups, and may require more complex models and reasoning. Direct target group identification implicitly solves hate speech classification, and this would likely propagate the error down-stream, likely making the results of our experiments less interpretable.

For further work, one could investigate directly identifying target groups, but this was deemed out of scope for the project. As a pointer to this experiment, a reasonable approach would be to compare direct target group identification versus first identifying hate-speech and target groups down-stream. In the subsection of 5.3.3 (*Natural Language Inference Transfer and Augmentation*), we highlight that there is reason to assume that making the model aware of hate-speech.

To exemplify the difference in difficulty between the tasks, we could even hypothesize that a method like simple dictionary look-ups could be used in identifying the target group, if we already assume hate-speech. This would, however, require hand-crafted expert-informed decision rules. Thus, using a machine learning approach is still merited, because it provides an accurate and scaleable approach, e.g. if the target groups were to change at some point.

Additionally, by explicitly setting the target groups, we impose an inductive bias into our model, which makes the problem easier, i.e. compared to an unsupervised-fashion, where the granularity of the target group is not predefined. This also falls inline with the results we observed, since the target group identification performs better than the hate-speech classifier in terms of raw performance metrics, as seen in figure 4.6.

Since we treated this as two separate tasks, we can see that efforts likely should be directed towards improving the hate-speech classification for better down-stream performance in the target group identification.

5.3.3 Natural Language Inference for Target Groups

All the transformer-based architectures for identifying the target groups were based on natural language inference. Another reasonable approach to solve this task, which still respects the multi-label setting, would be to train a classifier for each of the target groups. In a standard NLI approach, where we use all negative samples, each target group induces a data point for each premise. In terms of the number of gradient updates, one epoch in this model is equivalent to performing a gradient update in all of the individual classifiers.

We note that the main benefit of using an NLI approach is the fact that all inference is performed on a single model. Hence, we would only need to store the weights of one model (versus ten), and it may also be easier to use or deploy for the end-user. For further work, one could investigate how an NLI approach differs in performance between multiple classifiers. The single classifiers may perform better in terms of performance metrics, but this would have to be evaluated empirically. On the contrary, however, one could also hypothesize that a single model would allow for interactions and co-adaptations between the different target groups.

Natural Language Inference Transfer and Augmentation

Beyond the standard way of creating (`hypothesis`, `premise`)-pairs for the target groups, we also utilized information about e.g. toxicity and hate-speech only in training. In particular, we see that this approach greatly benefits target groups, where the number of training examples are sparse. We can view this as imposing an inductive bias into the model, which makes the model aware of concepts related to hatespeech. By including this into the training process, we could possibly attribute the increased performance to a transfer between tasks. This may also indicate that there are still a lot of potential performance gains by collecting more data for the scarce target groups.

Using NLI for classification is loosely reminiscent of data augmentation in image classification [8], since we can reeuse our data set to generate more data. However, the transformations applied in data augmentation for image classification typically enforce some kind of meaningful invariance with respect to e.g. scale, rotation, shift, etc.

For future work, one could consider how to construct the (`hypothesis`, `premise`)-pairs for classification, such that it enforces some kind of invariance, or more efficiently extracts implicit information from the data. However, as in the regular NLI case, this extra generated data may no longer be considered i.i.d., and we should not reasonably expect performance gains akin to simply obtaining more data. Thus, when training the model using these kinds of approaches, one should also consider whether the bottleneck is compute power or scarcity of data.

5.4 Countering hate speech and toxicity

Automatic toxicity removal

With the rise of LLMs, we wanted to find out whether it was possible to prompt an LLM to convert a comment which contains hate speech and toxicity into a version which does not contain any of it. For this we used the open source Llama 2 13B Chat model [19] . Since the comments we used for this experiment were in German, we prompted the model in German.

However, asking the model to change the comment into one without hate speech caused major issues when it contained toxic language and racial slurs. This was primarily caused by the alignment and bias, which censors the model and causes the model to output a predefined answer with the message that humans should be respectful to each other.

This issue could be bypassed by using the prompt: `"Wie würde der Kommentar aussehen, wenn dieser von einer nicht-rassistischer deutschen Person abgeändert wurde, so dass er nicht mehr beleidigend, diskriminierend, und rassistisch ist?"`. While the model reliably attempted to create comments which were less toxic and hateful, the suggestions it provided were mediocre at best. For example, a comment containing toxic language towards women was "corrected" by appending the phrase: `"Aber es ist wichtig, dass wir Menschen unabhängig von ihrem Beruf respektieren."`. Other comments were not corrected at all or on the other hand rewritten to a point that the original message was lost. Sometimes Llama 2 would just start outputting English text halfway through the response.

Even though the suggestions provided by Llama 2 were unsatisfactory, we still believe that there is potential for LLM-based systems to be capable of removing the hateful and toxic part of comments by using larger and uncensored LLMs (e.g. [18]). However, while this is possible for partially toxic and hateful comments, this is and will never be possible for comments containing solely hate speech and toxicity.

Integrated Gradients

As we have seen, Integrated Gradients can successfully highlight the most problematic part of a sentence and so give the user direct feedback on why their comment was removed and hopefully encourage a positive change. In combination with counter-speech that is specifically relevant to the target group found by the model it could be used as a proactive approach used to challenge and mitigate the effects of hate speech.

Chapter 6

Conclusion

Our project assessed various transformer-based models for identifying hate speech. We observed that larger models and datasets enhanced performance. Additionally, LoRA-based approaches cut training time in half while maintaining 95% efficacy. The fully fine-tuned E5 model performed the best in detecting hate speech and pinpointing the target group. Furthermore, this model outperformed all other models in accurately classifying target groups across all categories. We also found promising methods to proactively challenge and mitigate the effects of hate speech by using Integrated Gradients or Llama.

Future work suggests potential for further performance gains by scaling up the model and data, despite higher computational demands.

Appendix A

Hate Speech Classification

Table A.1: F1, Precision, Recall Scores for the Test set of models fine-tuned with LoRA

model_name	optimizer	seed	f1	precision	recall
bert_lora	adamw	42	0.579073	0.801292	0.453348
		43	0.591068	0.809774	0.465377
		44	0.585504	0.806408	0.459603
e5_lora	adamw	42	0.607126	0.851104	0.471861
		43	0.617452	0.859720	0.481708
		44	0.614275	0.854470	0.479489
	sophia	42	0.596633	0.877760	0.451899
		43	0.597085	0.887991	0.449748
		44	0.600555	0.874798	0.457219
e5_lora_nli	adamw	42	0.614778	0.849623	0.481645
		43	0.622571	0.854066	0.489809
		44	0.614122	0.855950	0.478837
	sophia	42	0.599214	0.872644	0.456254
		43	0.602199	0.884895	0.456395
		44	0.599289	0.874125	0.455937
e5_lora_nli_dual	adamw	42	0.613367	0.850027	0.479787
		43	0.617598	0.864163	0.480500
		44	0.615974	0.855547	0.481221
	sophia	42	0.605802	0.861605	0.467119
		43	0.611648	0.869548	0.471735
		44	0.608298	0.867528	0.468348
e5_lora_nli_dual+	adamw	42	0.610932	0.853931	0.475594
		43	0.617968	0.864432	0.480866
		44	0.613502	0.853931	0.478717
	sophia	42	0.609096	0.846527	0.475679
		43	0.618187	0.856624	0.483584
		44	0.616001	0.838584	0.486793
roberta_lora	adamw	42	0.600770	0.840334	0.467496
		43	0.605575	0.833603	0.475503
		44	0.608430	0.829833	0.480287

Table A.2: F1, Precision, Recall Scores for the Expert set of models fine-tuned with LoRA

model_name	optimizer	seed	f1	precision	recall
bert_lora	adamw	42	0.469636	0.391892	0.585859
		43	0.497959	0.412162	0.628866
		44	0.435484	0.364865	0.540000
e5_lora	adamw	42	0.636678	0.621622	0.652482
		43	0.615385	0.594595	0.637681
		44	0.597222	0.581081	0.614286
	sophia	42	0.662162	0.662162	0.662162
		43	0.651163	0.662162	0.640523
		44	0.660066	0.675676	0.645161
e5_lora_nli	adamw	42	0.641379	0.628378	0.654930
		43	0.645833	0.628378	0.664286
		44	0.642384	0.655405	0.629870
	sophia	42	0.658385	0.716216	0.609195
		43	0.679245	0.729730	0.635294
		44	0.637224	0.682432	0.597633
e5_lora_nli_dual	adamw	42	0.639456	0.635135	0.643836
		43	0.655518	0.662162	0.649007
		44	0.651007	0.655405	0.646667
	sophia	42	0.646667	0.655405	0.638158
		43	0.644518	0.655405	0.633987
		44	0.649007	0.662162	0.636364
e5_lora_nli_dual+	adamw	42	0.662252	0.675676	0.649351
		43	0.653199	0.655405	0.651007
		44	0.636364	0.662162	0.612500
	sophia	42	0.666667	0.668919	0.664430
		43	0.634483	0.621622	0.647887
		44	0.629758	0.614865	0.645390
roberta_lora	adamw	42	0.634146	0.614865	0.654676
		43	0.600000	0.567568	0.636364
		44	0.574545	0.533784	0.622047

Table A.3: F1, Precision, Recall Scores for the Test set of fully fine-tuned models

model_name	optimizer	seed	f1	precision	recall
bert	adamw	42	0.592491	0.816909	0.464803
		43	0.599176	0.822698	0.471164
		44	0.596408	0.815967	0.469954
e5	adamw	42	0.618088	0.850162	0.485545
		43	0.628315	0.846931	0.499405
		44	0.622439	0.840603	0.494183
	sophia	42	0.632352	0.799677	0.522933
		43	0.642939	0.797658	0.538490
		44	0.634481	0.789984	0.530129
e5_nli	adamw	42	0.625491	0.813947	0.507897
		43	0.634637	0.823371	0.516292
		44	0.625919	0.814082	0.508408
	sophia	42	0.630641	0.685784	0.583706
		43	0.642003	0.697361	0.594787
		44	0.633486	0.687130	0.587612
e5_nli_dual	adamw	42	0.628613	0.819871	0.509709
		43	0.628911	0.825256	0.508039
		44	0.630393	0.825121	0.510027
	sophia	42	0.633966	0.681206	0.592853
		43	0.642827	0.706651	0.589577
		44	0.633883	0.688072	0.587606
e5_nli_dual+	adamw	42	0.631123	0.814351	0.515203
		43	0.638833	0.813678	0.525840
		44	0.629342	0.813409	0.513208
	sophia	42	0.627802	0.646607	0.610060
		43	0.634401	0.646742	0.622522
		44	0.629465	0.649973	0.610212
e5_no_train	None	42	0.197960	0.176360	0.225590
		43	0.183416	0.199515	0.169721
		44	0.302400	1.000000	0.178134
roberta	adamw	42	0.614005	0.843430	0.482703
		43	0.622881	0.848411	0.492075
		44	0.617351	0.842084	0.487301

Table A.4: F1, Precision, Recall Scores for the Expert set of fully fine-tuned models

model_name	optimizer	seed	f1	precision	recall
bert	adamw	42	0.531250	0.459459	0.629630
		43	0.521073	0.459459	0.601770
		44	0.505837	0.439189	0.596330
e5	adamw	42	0.677419	0.709459	0.648148
		43	0.664430	0.668919	0.660000
		44	0.646259	0.641892	0.650685
	sophia	42	0.645161	0.608108	0.687023
		43	0.631206	0.601351	0.664179
		44	0.640569	0.608108	0.676692
e5_nli	adamw	42	0.631922	0.655405	0.610063
		43	0.644068	0.641892	0.646259
		44	0.610738	0.614865	0.606667
	sophia	42	0.569231	0.500000	0.660714
		43	0.550000	0.445946	0.717391
		44	0.541667	0.439189	0.706522
e5_nli_dual	adamw	42	0.626667	0.635135	0.618421
		43	0.623377	0.648649	0.600000
		44	0.619529	0.621622	0.617450
	sophia	42	0.566802	0.472973	0.707071
		43	0.544000	0.459459	0.666667
		44	0.560669	0.452703	0.736264
e5_nli_dual+	adamw	42	0.615385	0.621622	0.609272
		43	0.629758	0.614865	0.645390
		44	0.583893	0.587838	0.580000
	sophia	42	0.468468	0.351351	0.702703
		43	0.510823	0.398649	0.710843
		44	0.470085	0.371622	0.639535
e5_no_train	None	42	0.207921	0.141892	0.388889
		43	0.273438	0.236486	0.324074
		44	0.456790	1.000000	0.296000
roberta	adamw	42	0.637288	0.635135	0.639456
		43	0.616949	0.614865	0.619048
		44	0.613793	0.601351	0.626761

Appendix B

Target Group Identification

B.1 Natural Language Inference Example

Entailment: The first row demonstrates an entailment. Here, the premise is that reducing carbon emissions is crucial for slowing down global warming. The hypothesis states that lowering greenhouse gases will help combat climate change. The relationship is labeled as 'entailment' because if the premise is true (reducing carbon emissions is crucial for slowing global warming), it logically follows that the hypothesis is also true (lowering greenhouse gases helps combat climate change).

Contradiction: The second row shows a contradiction. The premise states that regular exercise is beneficial for maintaining good health, while the hypothesis asserts that being physically active has no impact on health. These two statements are in direct opposition; if the premise is true, the hypothesis must be false, leading to the label 'contradiction'.

Neutral:: The final row illustrates a neutral relationship. The premise is that reading books regularly can expand knowledge and improve vocabulary. The hypothesis is that reading science fiction novels is the best way to relax. In this case, the truth of the premise does not necessarily affirm or contradict the hypothesis. The two statements are related but do not have a direct logical inference, resulting in a 'neutral' classification.

B.2 Results for Target Group Identification

Table B.1: F1, Precision, and Recall Scores for 'gender'

model_name	optimizer	seed	precision	recall	F1 binary
deberta	adamw	42	0.7768595	0.893	0.68745189
		43	0.77685226	0.91434469	0.67530488
		44	0.76844784	0.9188641	0.66034985
deberta_lora	adamw	42	0.75959418	0.88946281	0.66281755
		43	0.75781948	0.91576674	0.64634146
		44	0.75064488	0.91509434	0.63629738
e5	adamw	42	0.76970228	0.89238579	0.67667436
		43	0.80770878	0.92179863	0.71875
		44	0.78328554	0.89429373	0.696793
e5_lora	adamw	42	0.74741108	0.90021692	0.63895304
		43	0.7530474	0.92358804	0.63567073
		44	0.74412533	0.92332613	0.62317784
e5_lora_nli_dual	adamw	42	0.8203125	0.76137113	0.8891455
		43	0.83110486	0.77189542	0.90015244
		44	0.83240223	0.79892761	0.86880466
e5_nli_dual	adamw	42	0.828125	0.76862228	0.89761355
		43	0.84656845	0.80081577	0.89786585
		44	0.83518006	0.79551451	0.87900875

Table B.2: F1, Precision, and Recall Scores for 'age'

model_name	optimizer	seed	precision	recall	F1 binary
deberta	adamw	42	0.46822742	0.82352941	0.3271028
		43	0.47557003	0.79347826	0.33953488
		44	0.46621622	0.8313253	0.32394366
deberta_lora	adamw	42	0.3772242	0.79104478	0.24766355
		43	0.37192982	0.75714286	0.24651163
		44	0.38571429	0.80597015	0.25352113
e5	adamw	42	0.50657895	0.85555556	0.35981308
		43	0.52037618	0.79807692	0.38604651
		44	0.47435897	0.74747475	0.34741784
e5_lora	adamw	42	0.39855072	0.88709677	0.25700935
		43	0.37226277	0.86440678	0.2372093
		44	0.36764706	0.84745763	0.23474178
e5_lora_nli_dual	adamw	42	0.5990099	0.63684211	0.56542056
		43	0.65747126	0.65	0.66511628
		44	0.56857855	0.60638298	0.53521127
e5_nli_dual	adamw	42	0.62745098	0.65979381	0.59813084
		43	0.67573696	0.65929204	0.69302326
		44	0.59410431	0.5745614	0.61502347

Table B.3: F1, Precision, and Recall Scores for 'sexuality'

model_name	optimizer	seed	precision	recall	F1 binary
deberta	adamw	42	0.60586319	0.88571429	0.46039604
		43	0.62271062	0.86734694	0.48571429
		44	0.56478405	0.86734694	0.41871921
deberta_lora	adamw	42	0.6038961	0.87735849	0.46039604
		43	0.59689922	0.92771084	0.44
		44	0.55892256	0.88297872	0.408867
e5	adamw	42	0.62745098	0.92307692	0.47524752
		43	0.61764706	0.86597938	0.48
		44	0.61146497	0.86486486	0.4729064
e5_lora	adamw	42	0.51929825	0.89156627	0.36633663
		43	0.57692308	0.88235294	0.42857143
		44	0.53103448	0.88505747	0.37931034
e5_lora_nli_dual	adamw	42	0.64139942	0.78014184	0.54455446
		43	0.64102564	0.72992701	0.57142857
		44	0.63126844	0.78676471	0.5270936
e5_nli_dual	adamw	42	0.66472303	0.80851064	0.56435644
		43	0.67092652	0.76086957	0.6
		44	0.64516129	0.79710145	0.54187192

Table B.4: F1, Precision, and Recall Scores for 'religion'

model_name	optimizer	seed	precision	recall	F1 binary
deberta	adamw	42	0.64339152	0.86577181	0.51190476
		43	0.60326087	0.81617647	0.47844828
		44	0.59620596	0.89430894	0.44715447
deberta_lora	adamw	42	0.58823529	0.82733813	0.45634921
		43	0.59945504	0.81481481	0.47413793
		44	0.56830601	0.86666667	0.42276423
e5	adamw	42	0.64321608	0.87671233	0.50793651
		43	0.62663185	0.79470199	0.51724138
		44	0.63819095	0.83552632	0.51626016
e5_lora	adamw	42	0.56233422	0.848	0.42063492
		43	0.51343284	0.83495146	0.37068966
		44	0.55462185	0.89189189	0.40243902
e5_lora_nli_dual	adamw	42	0.7443609	0.70714286	0.78571429
		43	0.70981211	0.68825911	0.73275862
		44	0.7394636	0.69927536	0.78455285
e5_nli_dual	adamw	42	0.75645756	0.70689655	0.81349206
		43	0.72164948	0.6916996	0.75431034
		44	0.72932331	0.67832168	0.78861789

Table B.5: F1, Precision, and Recall Scores for 'nationality'

model_name	optimizer	seed	precision	recall	F1 binary
deberta	adamw	42	0.82825607	0.93055556	0.74622116
		43	0.83028721	0.9244186	0.7535545
		44	0.81547354	0.92907801	0.72662441
deberta_lora	adamw	42	0.81455032	0.92780885	0.72593477
		43	0.82934262	0.92386033	0.75236967
		44	0.80717489	0.92975207	0.71315372
e5	adamw	42	0.84119053	0.92675921	0.77008751
		43	0.85005258	0.9091318	0.79818325
		44	0.83387481	0.91973244	0.76267829
e5_lora	adamw	42	0.80587571	0.93301936	0.70922832
		43	0.82882096	0.92675781	0.74960506
		44	0.80666217	0.9338197	0.70998415
e5_lora_nli_dual	adamw	42	0.86959847	0.83726068	0.90453461
		43	0.86514286	0.83554084	0.89691943
		44	0.86148008	0.82665696	0.89936609
e5_nli_dual	adamw	42	0.87334092	0.83442029	0.91607001
		43	0.86953243	0.83189033	0.9107425
		44	0.86486486	0.82688833	0.90649762

Table B.6: F1, Precision, and Recall Scores for 'disability'

model_name	optimizer	seed	precision	recall	F1 binary
deberta	adamw	42	0.27807487	0.76470588	0.16993464
		43	0.2513089	0.77419355	0.15
		44	0.21965318	0.76	0.12837838
deberta_lora	adamw	42	0.16374269	0.77777778	0.09150327
		43	0.14606742	0.72222222	0.08125
		44	0.06410256	0.625	0.03378378
e5	adamw	42	0.30687831	0.80555556	0.18954248
		43	0.34782609	0.76595745	0.225
		44	0.27027027	0.67567568	0.16891892
e5_lora	adamw	42	0.09815951	0.8	0.05228758
		43	0.14772727	0.8125	0.08125
		44	0.09937888	0.61538462	0.05405405
e5_lora_nli_dual	adamw	42	0.39183673	0.52173913	0.31372549
		43	0.34666667	0.6	0.24375
		44	0.31219512	0.56140351	0.21621622
e5_nli_dual	adamw	42	0.49446494	0.56779661	0.4379085
		43	0.43508772	0.496	0.3875
		44	0.45801527	0.52631579	0.40540541

Table B.7: F1, Precision, and Recall Scores for 'social status'

model_name	optimizer	seed	precision	recall	F1 binary
deberta	adamw	42	0.42112211	0.85983827	0.27884615
		43	0.41280654	0.84401114	0.27321912
		44	0.37933379	0.90291262	0.24010327
deberta_lora	adamw	42	0.36363636	0.88215488	0.22902098
		43	0.36363636	0.88194444	0.22903517
		44	0.33262562	0.93625498	0.20223752
e5	adamw	42	0.47355164	0.84684685	0.32867133
		43	0.47300771	0.82326622	0.33183048
		44	0.44730077	0.88324873	0.29948365
e5_lora	adamw	42	0.3460452	0.90073529	0.21416084
		43	0.36480687	0.88235294	0.22993688
		44	0.32932862	0.92094862	0.20051635
e5_lora_nli_dual	adamw	42	0.57197882	0.63665595	0.51923077
		43	0.54755043	0.58581706	0.51397656
		44	0.54571293	0.60695469	0.49569707
e5_nli_dual	adamw	42	0.58752166	0.58247423	0.59265734
		43	0.5648785	0.57462687	0.55545537
		44	0.58584071	0.60291439	0.5697074

Table B.8: F1, Precision, and Recall Scores for 'politics'

model_name	optimizer	seed	precision	recall	F1 binary
deberta	adamw	42	0.77446451	0.91878426	0.66932849
		43	0.77662503	0.93333333	0.66497462
		44	0.75237684	0.92165167	0.63563344
deberta_lora	adamw	42	0.76614794	0.92820248	0.6522686
		43	0.76877005	0.93740219	0.6515591
		44	0.74009196	0.92450766	0.61701351
e5	adamw	42	0.79229654	0.92237223	0.69437387
		43	0.789801	0.92207164	0.69071791
		44	0.7790795	0.91229789	0.67981015
e5_lora	adamw	42	0.75394936	0.9335477	0.6323049
		43	0.75573841	0.93817204	0.63270486
		44	0.74451273	0.93340671	0.61920409
e5_lora_nli_dual	adamw	42	0.82419786	0.76370393	0.89509982
		43	0.8326572	0.779924	0.89303843
		44	0.82209222	0.77741621	0.87221614
e5_nli_dual	adamw	42	0.82549774	0.76567349	0.89546279
		43	0.83601071	0.77625855	0.90572879
		44	0.82549317	0.76691729	0.89375685

Table B.9: F1, Precision, and Recall Scores for 'appearance'

model_name	optimizer	seed	precision	recall	F1 binary
		42	0.46153846	0.796875	0.32484076
deberta	adamw	43	0.41269841	0.79824561	0.27828746
		44	0.42926829	0.85436893	0.28664495
		42	0.375	0.76470588	0.24840764
deberta_lora	adamw	43	0.33004926	0.84810127	0.20489297
		44	0.35294118	0.82142857	0.2247557
		42	0.49676026	0.77181208	0.36624204
e5	adamw	43	0.46799117	0.84126984	0.32415902
		44	0.48868778	0.8	0.35179153
		42	0.38141809	0.82105263	0.24840764
e5_lora	adamw	43	0.30272953	0.80263158	0.18654434
		44	0.34936709	0.78409091	0.2247557
		42	0.57935285	0.56119403	0.59872611
e5_lora_nli_dual	adamw	43	0.55276382	0.61111111	0.50458716
		44	0.57471264	0.5794702	0.57003257
		42	0.59821429	0.56145251	0.64012739
e5_nli_dual	adamw	43	0.59304085	0.58682635	0.59938838
		44	0.60927152	0.61952862	0.59934853

Table B.10: F1, Precision, and Recall Scores for 'others'

model_name	optimizer	seed	precision	recall	F1 binary
		42	0.25744934	0.76595745	0.15472779
deberta	adamw	43	0.20792079	0.77777778	0.12
		44	0.191052	0.80612245	0.10836763
		42	0.10427807	0.78	0.05587393
deberta_lora	adamw	43	0.15938303	0.79487179	0.08857143
		44	0.09819121	0.84444444	0.0521262
		42	0.35409836	0.74654378	0.23209169
e5	adamw	43	0.35667396	0.76168224	0.23285714
		44	0.36613757	0.80092593	0.23731139
		42	0.2022756	0.76370393	0.11461318
e5_lora	adamw	43	0.18043202	0.779924	0.10142857
		44	0.18581907	0.76691729	0.1042524
		42	0.52218935	0.53975535	0.50573066
e5_lora_nli_dual	adamw	43	0.52243126	0.52932551	0.51571429
		44	0.54571226	0.56451613	0.52812071
		42	0.55163728	0.49213483	0.62750716
e5_nli_dual	adamw	43	0.54857898	0.5104551	0.59285714
		44	0.58925017	0.57069409	0.6090535

Appendix C

Hyperparameters

All our models were trained on a single node with 4 NVIDIA Tesla V100 GPUs. We use PyTorch Lightning [14] with DeepSpeed [3] to train our models on multiple gpus with mixed precision. We use a local batch size of 16, leading to a global batch size of 64.

Table C.1: Optimizer Hyperparameters

	optimizer	learning_rate	weight_decay
Full Finetuning	AdamW	5.0e-6	0.01
	Sophia	2.5e-6	0.10
LoRA	AdamW	3.0e-4	0.01
	Sophia	1.5e-4	0.10

Bibliography

[1] Xiangning Chen, Chen Liang, Da Huang, Esteban Real, Kaiyuan Wang, Yao Liu, Hieu Pham, Xuanyi Dong, Thang Luong, Cho-Jui Hsieh, Yifeng Lu, and Quoc V. Le. Symbolic discovery of optimization algorithms, 2023.

[2] Human Rights Council. Joint open letter on concerns about the global increase in hate speech. 2019.

[3] DeepSpeed Development Team. Deepspeed documentation, 2023. Accessed: 2023-12-18.

[4] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *North American Chapter of the Association for Computational Linguistics*, 2019.

[5] Ayako Hatano. Regulating online hate speech through the prism of human rights law: The potential of localised content moderation. *The Australian Year Book of International Law Online*, 41:127–156, 10 2023.

[6] Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *CoRR*, abs/2106.09685, 2021.

[7] Ana Kotarcic, Dominik Hangartner, Fabrizio Gilardi, Selina Kurer, and Karsten Donnay. Human-in-the-loop hate speech classification in a multilingual context, 2023.

[8] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*, pages 1097–1105. Curran Associates, Inc., 2012.

[9] Hong Liu, Zhiyuan Li, David Hall, Percy Liang, and Tengyu Ma. Sophia: A scalable stochastic second-order optimizer for language model pre-training, 2023.

[10] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692, 2019.

[11] Ilya Loshchilov and Frank Hutter. Fixing weight decay regularization in adam. *CoRR*, abs/1711.05101, 2017.

[12] United Nations. United nations strategy and plan of action on hate speech. 9 2019.

[13] Parliament of the United Kingdom. *Online Safety Act 2023*. 2023.

[14] PyTorch Lightning Team. Pytorch lightning documentation, 2023. Accessed: 2023-12-18.

[15] Jason D. M. Rennie, Lawrence Shih, Jaime Teevan, and David R. Karger. Tackling the poor assumptions of naive bayes text classifiers. In *Proceedings of the Twentieth International Conference on International Conference on Machine Learning*, ICML'03, page 616–623. AAAI Press, 2003.

[16] Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic attribution for deep networks. *CoRR*, abs/1703.01365, 2017.

[17] The European Commission. *Code of conduct on countering illegal hate speech online*. The European Commission, 2016.

[18] TheBloke. Dolphin 2.5 mixtral 8x7b gptq model. https://huggingface.co/TheBloke/dolphin-2.5-mixtral-8x7b-GPTQ, 2023. Accessed on 2023-12-19.

[19] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan

Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. Llama 2: Open foundation and fine-tuned chat models, 2023.

[20] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2023.

[21] Liang Wang, Nan Yang, Xiaolong Huang, Binxing Jiao, Linjun Yang, Daxin Jiang, Rangan Majumder, and Furu Wei. Text embeddings by weakly-supervised contrastive pre-training, 2022.

[22] Womankind Worldwide. Intersectionality 101: What is it and why is it important? https://www.womankind.org.uk/intersectionality-101-what-is-it-and-why-is-it-important/, 2023. Accessed: December 14, 2023.