How Fine-Tuned Encoder Models and a Zero-Shot Decoder Model Handle Annotator Disagreement in Hate Speech Detection

Ikraan Hayd (s3292320)

Supervisors:
Dr. Flor Miriam Plaza Del Arco & Dr. Lifeng Han

Bachelor Thesis– Informatica

Leiden Institute of Advanced Computer Science (LIACS)
liacs.leidenuniv.nl

August 1, 2026

Abstract

This study examines how fine-tuned encoder models and a zero-shot decoder model handle annotator disagreement in hate speech detection. Automated hate speech detection relies on labeled data, but human annotators do not always agree on what counts as hate speech. Most models are trained on majority labels without accounting for that uncertainty, which raises questions about how well they perform on genuinely ambiguous content. To investigate this, the HateXplain dataset was split into a high agreement subset and a low agreement subset, and five models were evaluated on each: BERT, HateBERT, RoBERTa, ToxicBERT, and Llama 3.1 70B. The first four were fine-tuned on labeled data, while Llama 3.1 was used in a zero-shot setting. All models performed better on the high agreement subset than on the low agreement subset, and the gap between the fine-tuned encoder models and Llama 3.1 narrowed on ambiguous content. The offensive class was the hardest to classify across most models and splits. Llama 3.1 showed a strong tendency to over-predict harmful labels, most often predicting hate speech for offensive content and offensive for normal content, while the fine-tuned encoder models missed more hate speech on ambiguous content. These errors were not evenly distributed across target communities: the encoder models flagged normal content concerning race and religion as harmful more often, while missing hate speech concerning gender and sexuality more often. These findings show that model errors are directly tied to annotator disagreement, and that automated hate speech detection is least reliable exactly where the content is hardest to judge, and least fair exactly where a target group already has the least protection.

Contents

1 Introduction **1**
- 1.1 Research questions . . . 2
  - 1.1.1 Sub-questions . . . 2
- 1.2 Thesis Overview . . . 2

2 Background **4**
- 2.1 What is Hate Speech . . . 4
- 2.2 NLP for hate speech detection . . . 4
  - 2.2.1 Transformer-based models . . . 5
- 2.3 Large Language Models in Hate Speech Detection . . . 5
- 2.4 HateXPlain . . . 6
  - 2.4.1 Annotator disagreement . . . 6
- 2.5 Ethical Implications of Automated Hate Speech Detection . . . 7
- 2.6 Related Work . . . 7

3 Approach **10**
- 3.1 Annotator Agreement Split . . . 10
- 3.2 Models . . . 10
- 3.3 Data Processing . . . 11
- 3.4 Evaluation . . . 13

4 Results **15**
- 4.1 Model Performance Across Dataset Splits . . . 15
- 4.2 Error Patterns and Misclassifications . . . 19
- 4.3 Statistical Significance . . . 19
- 4.4 False Positive rate on True-Normal content . . . 21
- 4.5 False Negative Rate on True-Hate-Speech Content . . . 23
- 4.6 Qualitative Examples of Misclassification . . . 24
- 4.7 Training Convergence . . . 25

5 Discussion **28**
- 5.1 Ambiguity and Model Performance . . . 28
- 5.2 The Difficulty of the Offensive Class . . . 28
- 5.3 Divergent Error Patterns and Ethical Risks . . . 29
- 5.4 Disproportionate Impact on Marginalised Communities . . . 29
- 5.5 Limitations . . . 30

6 Conclusion **32**

References **35**

A Confusion Matrices **36**
A.1 Full Dataset 36
A.2 High Dataset 37
A.3 Low Dataset 38

1 Introduction

The Challenge of Automated Content Moderation

Social media platforms process millions of posts every day, and a significant portion of that content is hateful or harmful. Moderating it all by hand is simply not possible at that scale, which has led to a growing reliance on automated hate speech detection systems [FBP+20]. These systems are built on machine learning models trained to classify content as hate speech, offensive, or normal [MZ21].

Transformer-based models such as BERT and RoBERTa have become the standard approach for this task, and more recently large language models have entered the picture as well [AID+25]. While these models perform well overall, research has shown that they struggle with ambiguous content and can produce errors that disproportionately affect already marginalised communities [SCG+19, MYY+19]. A key source of this ambiguity is annotator disagreement: when human annotators cannot agree on how to label a post, the resulting label is uncertain, and models trained on that label inherit that uncertainty [DDP22]. Despite this, most studies treat majority labels as ground truth without examining what happens to model performance on the cases where humans themselves were not sure. This study addresses that gap.

To do so, the HateXplain dataset was split into a high agreement subset, where all three annotators agreed on the label, and a low agreement subset, where only two of three agreed. Five models were then evaluated on each subset: four encoder-only models (BERT, HateBERT, RoBERTa, and ToxicBERT) were fine-tuned on the labelled data, while a fifth, larger model (Llama 3.1 70B) was evaluated in a zero-shot setting, without any task-specific training. Comparing these two very different approaches, on both easy and hard content, makes it possible to see not only how well each model performs, but where and how their errors differ. To illustrate what this classification task actually involves, consider three short examples.

> ***Content warning:*** *the example below is an authentic post from the HateXplain dataset (sourced from Gab and Twitter) and contains an explicit racial slur alongside graphic, degrading language. It is included unedited because the offensive language itself is what the model must learn to recognise; readers who would prefer to skip it can continue at Section 1.1.*

Example E1 (target: race, true label: hate speech) — **BERT**: hate speech, **HateBERT**: hate speech, **ToxicBERT**: hate speech, **RoBERTa**: hate speech, **Llama**: hate speech.

> `"clearly the libtard ground wale found a nigger to screw her probably the only time in her life she ever got laid how do you fuck a fat woman leaf threw the folds of fat until you find one that smells like shit then turn back one"`

Example E2 (target: gender, true label: offensive) — **BERT**: offensive, **HateBERT**: offensive, **ToxicBERT**: offensive, **RoBERTa**: offensive, **Llama**: offensive.

> `"fucked a white bitch in the ass she shitted call it a mcflurry"`

Example E3 (target: none, true label: normal) — **BERT**: normal, **HateBERT**: normal, **ToxicBERT**: normal, **RoBERTa**: normal, **Llama**: normal.

```
"the asian men who can not find wives in their own countries could perhaps
be introduced to the american and european women who can not find husbands
in their own countries"
```

Across all three classes, the five models agree unanimously when a post is clearly hateful, clearly vulgar without targeting a group, or clearly unremarkable. As the remainder of this thesis shows, the real difficulty arises not on content like this, where annotators themselves agreed, but on the instances where human annotators could not agree. This thesis investigates that connection between annotator disagreement and model error, and examines the resulting false positives and false negatives through the ethical lens of who is silenced and who is left unprotected when such errors occur.

1.1 Research questions

How do the misclassifications of fine-tuned encoder models and a zero-shot decoder model correlate with annotator disagreement in the HateXplain dataset, and what are the ethical implications of automating decision-making in cases where human annotations are ambiguous?

1.1.1 Sub-questions

- RQ1: *How do encoder-only models BERT, HateBERT, ToxicBERT, and RoBERTa and the decoder-only model Llama 3.1 perform on the HateXplain dataset across instances with high versus low annotator agreement?*
- RQ2: *How does the performance of the fine-tuned encoder-only models compare to the zero-shot decoder-only model, and what does this reveal about how fine-tuning versus zero-shot prompting handles instances with low annotator agreement?*
- RQ3: *How do false positives and false negatives differ across models between instances with high and low annotator agreement, and what ethical risks does this reveal about automated hate speech detection in ambiguous cases?*

1.2 Thesis Overview

The remainder of this thesis is structured as follows. Section 2 provides the background needed to situate this study: it discusses how hate speech is defined in the literature, how NLP has been applied to its detection, the HateXplain dataset and the concept of annotator disagreement, the ethical implications of automated hate speech detection, and related work. Section 3 describes the approach taken to answer the research questions, covering how the dataset was split by annotator agreement, the five models used, the data processing pipeline, and the evaluation metrics. Section 4 presents the results: model performance across the three dataset splits, the error patterns visible in the confusion matrices and in the false positive and false negative rates by target community, the results of McNemar's test, concrete qualitative examples of misclassification, and the training and evaluation loss curves confirming that training converged correctly. Section 5 discusses what these results reveal about annotator disagreement, the offensive class, the divergent error patterns between the encoder-only models and Llama 3.1, and the disproportionate impact on marginalised

communities, before addressing the limitations of this study. Section 6 concludes the thesis by answering the research questions and suggesting directions for future work.

2 Background

This section provides the theoretical foundation for this study. It begins with an overview of how hate speech is defined in the literature Section 2.1, followed by a discussion of how Natural Language Processing has been applied to its automated detection, covering classical methods and transformer-based models, Section 2.2. Section 2.3 then discusses the use of large language models for hate speech detection. Section 2.4 introduces the HateXplain dataset and the concept of annotator disagreement, Section 2.5.discusses the ethical implications of automated hate speech detection, and the section closes with a discussion of related work in Section 2.6.

2.1 What is Hate Speech

No universal definition of hate speech exists in international human rights law to this day. The matter is one of ongoing debate, especially when it comes to questions of equality, non-discrimination, and freedom of expression [FN18] [DWMW17]. Some argue that the term is best understood as a collection of various expressions shared by common characteristics, which is why a one-size-fits-all definition is hard to come by [Bro17].

In the NLP literature, however, certain patterns emerge from the various definitions put forward. For the most part, hate speech is characterized as a deliberate act against a particular group, driven by either real or perceived traits of their identity [DGPGPC18]. A review of the field by Fortuna and Nunes (2018) demonstrated that the most common threads are a specific target and some form of attack or diminution [FN18]. Davidson et al. (2017) put it in terms of language meant to insult, humiliate, or disparage members of a target group, or to show them hatred [DWMW17]. Yin and Zubiaga (2021) are on the same page, defining it as content that stokes or directs hatred at a group on the basis of religion, ethnicity, or sexual orientation [YZ21].

The core point these definitions agree on is that hate speech is more than just unpleasant or offensive words. It is language with the express purpose of targeting people for who they are, whether that be their gender, religion, or ethnicity [DGPGPC18] [FN18] [DWMW17].

2.2 NLP for hate speech detection

With the sheer volume of material put up on social media on a daily basis, it is simply not possible to have someone go through and review it all for hate speech by hand [FBP+20]. For that reason, researchers have turned to Natural Language Processing (NLP) and cast the problem of hate speech detection as one of text classification. Here, machine learning models are put to work to automatically spot and put harmful content in its place [MZ21]. In practical terms, a model is given a piece of text and is trained to determine whether it is hate speech, offensive language, or neither [DWMW17].

The NLP approach to this usually begins with some housekeeping on the raw text: you strip out characters that don't matter, put the spelling in order and break the text into units the model can handle [MYY+19]. After the text has been tidied up, it is reduced to numbers. The older way of doing things was to employ something like TF-IDF to get a count of word frequency and run it through a Naive Bayes or SVM classifier [MYY+19]. While those are straightforward and easy to make sense of, they do have a flaw in that they treat words in isolation and cannot grasp context [MYY+19]. Newer methods have moved on from that to deep learning and word

embeddings which pick up on the semantics between words [FN18]. But the big thing to come along in recent times is transformer-based models, BERT being the prime example.

2.2.1 Transformer-based models

Transformer models have done much to advance the automatic detection of hate speech [MYY+19] [Kor21]. In their wake, a number of pre-trained language models have been put to wide use in text classification, all with their own way of making sense of context.

This development largely started with BERT (Bidirectional Encoder Representations from Transformers), the foundational model on which many others are built. What makes BERT unique is its approach to reading text. Where older models processed sentences word for word from left to right, BERT takes in the entire sentence and the surrounding context all at once [Kor21]. This allows the model to understand that a word can mean something quite different depending on how it is used. After being pre-trained on a large body of general text, BERT is fine-tuned for a specific task, such as hate speech detection, to put its learned knowledge to work [Kor21] [DWMW17].

RoBERTa is, in effect, a more thoroughly trained BERT. Liu et al. (2019) made the case that BERT was not being trained to its full potential [LOG+19]. They set about training on more data over a longer period and did away with a step that was actually doing more harm than good. The resulting Robustly Optimized BERT Pretraining Approach outperforms BERT on a host of tasks and yet has the same structure [LOG+19].

HateBERT was put forward by Caselli et al. (2021) on the premise that you should train a model on abusive language from the get-go rather than try to refine the process. They retrained BERT using a trove of Reddit comments from communities that had been banned for their offensive or hateful posts [CBMG21]. The logic was simple: if the model has been exposed to plenty of abuse in training, it ought to be better at spotting it. And indeed, across several datasets HateBERT has been shown to do a better job of detecting hate speech than standard BERT [CBMG21].

The Unitary team and Hanu (2020) have their ToxicBERT, which is fine-tuned for the purpose of finding toxic comments online [HT20]. Drawing on data from various Jigsaw challenges, it is capable of picking up on everything from insults and threats to identity-based hate. But what sets it apart is the effort to curb bias. Too often a classifier will wrongly link toxicity to neutral sentences simply because they mention a race or religion. ToxicBERT was made to put a stop to that kind of thing, no small matter when it comes to hate speech detection [HT20].

2.3 Large Language Models in Hate Speech Detection

The hate speech detection literature has seen the arrival of a new breed of model in the form of large language models (LLMs), moving on from the kind of fine-tuned transformers we have been looking at. Where models such as HateBERT and ToxicBERT are trained or fine-tuned on specific, labelled datasets, LLMs such as GPT-4 and LLaMA distinguish themselves through their exceptional scale and the breadth of their pre-training [AID+25].

With LLMs, little task-specific fine-tuning is needed to achieve results, given the breadth of corpora on which they have been pre-trained. Then there is their in-context learning, which is what you will not find in an encoder-only model; they can handle a classification job from the prompt alone, no parameter updates required [ZZL+23]. The model doesn't rely on being trained for specific labels but instead puts its pre-training to use on new tasks via some well-put-together

textual instructions [GDF25]. Even when it comes to something like flagging hateful language, be it overt or of a more subtle nature, a handful of good examples will be sufficient to guide the model. In fact, research has borne out that this kind of flexibility works just as well in multi-shot learning scenarios [AID+25] [CAG25].

2.4 HateXPlain

A text classification model cannot be trained or evaluated without labeled data. For automated hate speech detection, there are a number of benchmark datasets that have been put together for just that end; HateXplain is perhaps the best known of them.

But what sets HateXplain apart from older hate speech datasets is how its annotations are put in place. Instead of slapping a single categorical label on a post, each one is appraised from three angles [MSY+21].

Take the class label for instance: a post will be put in one of three boxes, normal, offensive or hate speech [MSY+21]. They made a point of keeping those categories distinct because mixing up hate speech and offensive language is a problem, as we discuss in Section 2.1.

Then there is the target community, the second dimension, which tells you the specific group the post is aimed at by way of race, gender, religion or sexual orientation.

The third dimension is the rationale and it is the most telling. The annotators had to zero in on the actual words or phrases in a post that led them to their decision [MSY+21], giving you more than a label and showing you precisely what in the text was the cause.

As for the data itself, it comes from Twitter and Gab, two staples of hate speech research [MSY+21]. After removing duplicates and reposts, the final dataset consisted of roughly 20,000 entries, including 11,093 from Gab and 9,055 from Twitter [MSY+21]. Three people from Amazon Mechanical Turk would independently label each post. If at least two of them were in agreement, that became the ground truth by way of majority vote. In the 919 cases where they could not come to an accord, the post was simply left out of the final dataset [MSY+21].

2.4.1 Annotator disagreement

Unanimous verdicts are unlikely when several people label the same text. To put a number on the consistency of those decisions, we look at inter-annotator agreement (IAA). If the IAA is high you can be sure the labels are reliable and the task was well defined; a low figure is a sign of ambiguity or that the annotators are seeing things differently.

Cohen's kappa is the standard metric for the job. It will tell you how much two annotators agree, factoring in the possibility of random chance [DDP22]. A 1 is as good as it gets, 0 means they are little more than guessing, and anything below zero points to a systematic disagreement. When more than two annotators are involved, Krippendorff's alpha is the preferred measure, as it can achieve the same result with multiple parties at once [MSY+21].

Take the HateXplain dataset for instance: its Krippendorff's alpha of 0.46 puts it in the moderate range and above most other hate speech datasets [MSY+21], but even so it is enough to give you an idea of the subjective nature of the work involved.

2.5 Ethical Implications of Automated Hate Speech Detection

There is real potential for a hate speech detection model to do damage when it misclassifies content. From a classification standpoint, two kinds of error must be considered [MYY+19]. One is the false positive: the model labels a post as hate speech even though there is nothing harmful about it. Then there is the false negative, in which the model simply does not pick up on a post that is overtly hateful. While both are errors of consequence, they don't inflict the same sort of harm [MYY+19].

With a false negative, the model has failed in its primary purpose: it lets harmful material remain on the site and the target of that content is left without protection. A false positive is different; here a post is taken down for no justifiable cause. It might be tempting to think of that as the lesser evil, but denying someone the right to put their thoughts out there online is no small matter, particularly for those in communities that are already short-changed. It robs them of a voice in the conversation [MYY+19]. Take African American English for instance: studies have found that models working from common hate speech data will all too often label a tweet as offensive when it is not [SCG+19]. Research has shown cases where almost half of entirely unremarkable tweets in that dialect were flagged as such. In the end, what was put in place to shield people can wind up harming the very ones the system was intended to help.

This can be put down to the fact that bias has a way of seeping in via the training data and the models. There is research to suggest that a model might give the impression of good performance yet be coming to its decisions for the wrong reasons [Els22]. A high accuracy score should not deceive one into thinking the model has any real grasp on hate speech; it could be as simple as latching onto some surface level patterns in the data that coincide with the labels [Els22]. The trouble is, such patterns are more likely to be a mirror of societal prejudices than an actual comprehension of why something is harmful.

Then there is the matter of social stereotypes. Research indicates that hate speech classifiers are capable of learning and putting them into practice [VCH+23]. When a model is put to work on data laced with biased assumptions about certain groups, it does more than take in those prejudices; it has the potential to amplify and propagate them. So deploying such models across social media is hardly a neutral act. In fact, as has been noted, it can be an active part of what makes existing inequalities worse [VCH+23] [Els22].

Putting the scale of these systems into perspective reveals why such matters are of paramount importance. An automated model for hate speech processes millions of posts a day, with little to no human oversight. If that model is biased or has trouble with something ambiguous, it is not an error confined to a few instances; they accumulate over an extraordinary volume of content [FDWT22] [MYY+19]. It can be argued that most conventional NLP techniques are ill suited to a task as nuanced and context driven as this one [FDWT22] [Els22]. There is no room to be casual about the ethics involved. Understanding where and why models go wrong, particularly on the cases that even humans find hard to agree on, is a necessary step toward building systems that are actually fair and reliable [FDWT22] [VCH+23] [DDP22].

2.6 Related Work

The early days of hate speech detection research were for the most part foundational, with the aim of putting together datasets and seeing what an automated model could be made to do. Davidson et al. (2017) were pioneers in this regard, being some of the first to put the difference between offensive

language and hate speech under a microscope [DWMW17]. To do so they built a lexicon of hate terms and used the Twitter API to pull in tweets; these were then handed off to crowdworkers to be sorted as either hate speech, offensive or neither. From the resulting data they derived TF-IDF features for a logistic regression classifier. While the model was serviceable enough, it had a habit of muddling hate speech with offensive language, something you would find the human annotators doing just as often. What one is left with from their study is less a lesson in methodology than in concept: the boundary between hate speech and offensive language is inherently ill defined and does not admit of neat categories.

In an attempt to establish a clearer picture of the field, Fortuna and Nunes (2018) surveyed the existing datasets, annotation practices and machine learning methods [FN18]. Time and again they encountered the same problems. For one thing, there is no consensus as to what constitutes hate speech, which makes it near impossible to compare datasets from different research groups. Furthermore, models proved more inclined to pick up on superficial word-level cues than to grasp the actual substance of a post. Crucially, however, Fortuna and Nunes (2018) argued that the fault does not lie with the limitations of the models alone; the data was defective to begin with [FN18].

There has been no shortage of change in NLP since the advent of transformers, and hate speech detection is one area that has not been left behind. In a 2019 paper, MacAvaney and colleagues compared a range of approaches, from classical methods to some of the first neural models [MYY+19]. What emerged from their study was an unambiguous finding: bias is a problem that cannot be ignored. The training data contained hateful content associated with certain demographic terms more than others, and the models were quick to associate those terms with a hateful label. The result was that a neutral post from a marginalised community could be wrongly identified as harmful. As MacAvaney et al. argued, there is little point in reporting strong accuracy figures if the model is causing harm in particular instances [MYY+19].

A more in depth examination by Sap and colleagues (2019) put the matter of race under a microscope [SCG+19]. Their work showed that an automated system will routinely flag a tweet as offensive if it is in African American English, no matter how unobjectionable the actual content may be. You could call it a structural problem rather than some oddity in the model. The fact is, if your training data has the social biases of the world in it, the model will reflect that. In the end you have a tool meant to offer protection being turned on the very people it should be for [SCG+19].

In order to address such issues, Mathew and colleagues (2021) put together the HateXplain dataset [MSY+21]. Previous datasets tended to offer a single label for any given post, whether from one annotator or by way of a majority vote. The new dataset was conceived to do more than that. As we cover in Section 2.4, they have provided annotations from three separate annotators as well as details on the target community and the particular words behind their reasoning. In this way you can see where there is consensus between annotators and not simply the end result. And if the annotators could not agree on a post, it was left in the dataset instead of being weeded out.

Davani et al. conducted this investigation in a 2022 paper [DDP22]. Their goal was to investigate the consequences of treating majority labels as the ground truth. They believe that areas where annotators cannot agree are not merely noise, but rather a manifestation of genuine ambiguity within the material. They were correct; when examining how models perform on disputed cases, it is evident that they are not comfortable with ambiguity, as their training has been based on the majority opinion [DDP22]. What Davani et al. found is that allowing a model to present the full range of an annotator's judgment, rather than a single label, leads to better results on difficult cases and provides a more accurate reflection of subjectivity. Ultimately, they contend that standard

pipeline configurations are discarding valuable signals that ought to be present [DDP22].

A distinct yet related issue concerns the bias inherent in the models themselves. Elsafoury [Els22] argued that there is no necessary link between a high benchmark score and true fairness. It is possible for a model to perform well on paper but be wrong for the wrong reasons; its errors tend to mirror and reinforce existing inequalities. Once millions of such decisions are made without human oversight, the damage accumulates rapidly [Els22].

Fortuna et al. [FDWT22] have done much to put these concerns in perspective. They make the case that the field's preoccupation with accuracy is itself an issue. The way hate speech is, for all its complexity, does not lend itself well to standard approaches; you need an evaluation process that can grapple with context and subjectivity as well as the impact on the ground. Then there is the matter of a model's failure: they would have us be more forthright about what that entails and who is left to foot the bill [FDWT22].

In a 2025 study, Dehghan et al. set out to address the problem of annotator disagreement in the training process [DSY25]. They put a number of methods to the test, from soft labeling to multi-annotator modeling, to see how they would hold up when there was varying degrees of agreement among the annotators. Their findings were that you get better generalization from models built with this kind of disagreement in mind, particularly when it comes to the more ambiguous posts [DSY25].

Combining these efforts demonstrates significant progress in detecting hate speech, although the most difficult cases remain. For the most part, research has followed one of two paths. Research into annotator disagreement helps explain why some labels are difficult to assign, whereas other studies examine the harm caused by model errors, such as innocent content being misflagged or hateful material being overlooked. It is rare for these two approaches to be combined directly. Muhumuza et al. (2026) took a step in this direction, showing that annotator disagreement in HateXplain concentrates specifically at the boundary between the hate speech and offensive classes, and that both hard-label and soft-label BERT models suffer a substantial accuracy drop on disagreement instances compared to agreed instances [MAA26]. Muhumuza et al. (2026) work does not look at the different kinds of errors a model can make, does not compare very different model types such as a fine-tuned encoder against a zero-shot decoder, and does not check whether these errors hit certain target communities harder than others. This changes how we should think about such errors: they are not just technical glitches, but the kind of judgment call a person would also struggle with. This study builds on that idea, looking across five models and different error types at whether the cases a model gets wrong are the same ones humans could not agree on, and what that means for the communities affected by those mistakes.

3 Approach

This Section describes the approach taken to answer the research questions outlined in Section 1.1. The dataset and the way it was split based on annotator agreement are discussed in Section 3.1. Section 3.2 covers the models selected for this study and the rationale behind their inclusion. Section 3.3 describes how the data was processed prior to training and evaluation, and Section 3.4 sets out the evaluation metrics and statistical tests used to compare model performance.

3.1 Annotator Agreement Split

To look at how annotator disagreement impacts model performance, the dataset was split into three distinct subsets based on how well the annotators agreed:

- The Full Dataset: Contains all available instances without any filtering, serves as baseline to measure overall classification performance.
- The High Agreement Set: Includes only the instances where all three annotators agreed on the same label.
- The Low Agreement Set: Contains instances where exactly two out of three annotators agreed.

Following that, each subset was split into a training set (80%) and a testing set (20%). To ensure that the results remain perfectly comparable across different models, a fixed random seed of 42 was used for all splits, guaranteeing that every model is trained and tested on the exact same data points. Table 1 reports the number of instances in each split.

Split	Total instances	Train (80%)	Test (20%)
Full	15,383	12,306	3,077
High agreement	7,888	6,310	1,578
Low agreement	7,495	5,996	1,499

Table 1: Number of instances per dataset split

3.2 Models

Five models were put to the test in this study: a large language model and four that are transformer-based. An account of their theoretical underpinnings can be found in Sections 2.2 and 2.3.

The transformers used were BERT, HateBERT, ToxicBERT and RoBERTa. Each of these was fine-tuned on the HateXplain dataset for three-class classification, loading the pretrained checkpoints from Hugging Face as follows: `bert-base-uncased` for BERT, `GroNLP/hateBERT` for HateBERT, `unitary/toxic-bert` for ToxicBERT and `cardiffnlp/twitter-roberta-base-hate` for RoBERTa. With the help of the Hugging Face Trainer API, each was given a classification head with three output labels (hate speech, offensive, normal). In the case of ToxicBERT an extra configuration change was needed; its architecture is built for multi-label work so the problem type had to be explicitly set to single-label to suit the task. To be sure any variation in performance

comes down to the model and not the training setup, all four were run with the same configuration, see Section 3.3 for details.

Then there is Llama 3.1 70B. The 70B variant was chosen over the 8B because the larger parameter count has been proven to give better reasoning and instruction-following [GDJ+24]. But unlike the transformers, it was not fine-tuned. Llama 3.1 70B was left to work in a zero-shot setting, classifying a post from a structured prompt alone with no labelled examples to go on. A model of that size is too much for a standard workstation, so the LIACS computing server was used and the model was deployed through Ollama's local inference interface. Having Llama 3.1 in the mix with the four fine-tuned transformers is what makes a direct comparison between prompting and supervised fine-tuning possible.

3.3 Data Processing

Before the data could be put to the test in training and assessment, we had to run it through some preprocessing. Take HateXplain for instance: rather than a post being one long string, we view it as an assemblage of tokens. We would reassemble those into a sentence with a space between words and then assign a ground truth label by way of the three annotators work, the post was given whichever of their labels was most common. The process used to arrive at those majority labels and to split the dataset is described in Section 3.1.

For the transformer quartet: RoBERTa, BERT, ToxicBERT and HateBERT, we needed to use their respective tokenizers to produce the input IDs, attention mask and token type IDs they require. To ensure uniformity we imposed a hard 128 token limit; any excess was cut off and the rest padded out. After that we did some housekeeping for the Hugging Face Trainer, turning the relevant columns into PyTorch tensors and jettisoning superfluous ones such as the original post tokens or rationale annotations.

All four encoder-only models were fine-tuned using an identical training configuration, ensuring that any difference in performance can be attributed to the models themselves rather than to differences in the training setup. The full set of hyperparameters is listed in Table 2. Epochs, batch size, evaluation strategy, and the random seed were set explicitly for every training run. The learning rate, optimizer, weight decay, and learning rate scheduler were not set explicitly and therefore correspond to the Hugging Face `TrainingArguments` defaults at the time of training (a learning rate of 5e-5, the AdamW optimizer, no weight decay, and a linear scheduler without warmup).

Hyperparameter	Value
Models fine-tuned	BERT, HateBERT, RoBERTa, ToxicBERT
Epochs	3
Train batch size	16
Eval batch size	16
Max sequence length	128 tokens
Learning rate	5e-5
Optimizer	AdamW
Weight decay	0.0
Learning rate scheduler	Linear, no warmup
Evaluation strategy	Per epoch
Model selection	Best checkpoint on eval loss, kept at end
Random seed	42

Table 2: Fine-tuning hyperparameters used for the four encoder-only models. All four models were trained with an identical configuration to isolate architecture effects.

Llama 3.1 is another story. There was no need to tokenize here, so we let the model have at it with every post fed in as an unprocessed prompt. As you will see in Listing 1, we set the prompt to instruct the model to file the post under one of three categories, laying out a few rules and what each means along the way. We asked for nothing more than the label in return and, with the temperature set to zero, we are assured of consistency from the model. The raw text output was then made into a numerical class: 0 for "hate", 1 for "offensive" and a default of 2 for normal. If the model failed to come up with an expected label, we count it as a miss on the offensive material.

```
prompt = (
    "Classify the following social media post into exactly one of these
       three categories:\n"
    "- hate speech\n"
    "- offensive\n"
    "- normal\n\n"
    "Rules:\n"
    "- 'hate speech' targets a person or group based on protected
       characteristics "
    "(race, religion, gender, sexuality, etc.)\n"
    "- 'offensive' contains rude or vulgar language but does not target a
        group\n"
    "- 'normal' is neither hateful nor offensive\n\n"
    ...
    "Respond with ONLY the label, nothing else. "
    "Choose from: hate speech, offensive, normal"
)
```

Listing 1: The prompt template used for the Llama 3.1 classification task.

3.4 Evaluation

Various standard classification metrics were used to evaluate the performance of each model. The metric relied on most is the macro-averaged F1-score, which is obtained by calculating the F1 score for each class and then averaging these values. This metric was prioritised because the three classes in HateXplain are not balanced, as hate speech is significantly underrepresented compared to offensive or normal content, as shown in Table 3. The macro average is appropriate here, as it assigns equal weight to each class regardless of how frequently it appears in the data.

Total accuracy, precision, and recall were likewise recorded. Precision indicates the percentage of predictions for a specific class that were correct, whereas recall reveals the proportion of actual instances of that class the model successfully identified. Accuracy is defined as the percentage of posts correctly categorised. To gain a clearer understanding of the types of errors a model commits, confusion matrices were generated for each dataset split and model. These matrices illustrate where one label has been incorrectly classified as another.

Then there is the question of whether any performance gap between the models is substantive or merely coincidental. To address this, McNemar's test [McN47] was applied to all pairs of models. This approach compares two models by analysing instances where one succeeded while the other failed, considering any result with a p-value below 0.05 as statistically significant. These assessments

were conducted separately for the entire dataset, as well as for the subsets with high and low agreement.

To examine whether model errors are distributed evenly across the target communities annotated in HateXplain, an additional analysis was performed on the saved model predictions. Each HateXplain instance is annotated by its three annotators with a target community for the post (e.g. race, gender, religion, sexuality, or none), in addition to the class label. For every test instance, the target community was determined by taking the most frequently mentioned community across the three annotators, following the same majority-vote principle used to determine the class label in Section 3.1; instances for which no annotator identified a specific target were labelled "None". These raw target categories were then grouped into six broader categories; race, gender, religion, sexuality, disability, and none, based on the target labels defined by Mathew et al. (2021) [MSY+21]. This analysis required no additional training: the target community of each instance was matched against the class predictions already produced by each of the five models (Section 3.2) on the same full, high, and low agreement test splits used throughout this study. For each model and target group, the false positive rate on the normal class was computed as the proportion of test instances with a true label of "normal" that the model misclassified as either hate speech or offensive.

Split	Hate speech	Offensive	Normal	Total
Full	956	885	1,236	3,077
High agreement	476	295	807	1,578
Low agreement	486	593	420	1,499

Table 3: Class distribution of the test set for each dataset split

4 Results

In this Section the results are presented across three dataset splits: the full dataset, the high agreement dataset, and the low agreement dataset, each summarised in a dedicated classification report containing precision, recall, and F1-score for all five models. Section 4.1 presents model performance across the three splits: the classification report for the full dataset is presented in Table 4, the high agreement subset in Table 5, and the low agreement subset in Table 6. Section 4.2 discusses the error patterns and misclassifications visible in the confusion matrices. Section 4.3 reports the results of McNemar's test comparing all model pairs across the three splits are reported in Table 7. Section 4.4 reports the false positive rate on true-normal content, broken down by target community, is reported in Table 8. Section 4.5 reports the false negative rate on true-hate-speech content, broken down by target community, is reported in Table 9. Section 4.6 presents concrete qualitative examples of misclassification that illustrate these quantitative patterns. Section 4.7 reports the training and evaluation loss curves for the four encoder-only models, confirming that training converged correctly. Confusion matrices for all models and dataset splits are provided in Appendix A.

4.1 Model Performance Across Dataset Splits

Table 4 shows the results for all five models on the full dataset. Among the encoder-only models, RoBERTa achieves the highest macro F1-score of 0.70, followed by BERT at 0.68, HateBERT at 0.67, and ToxicBERT at 0.65. All four models perform best on hate speech and normal content, while the offensive class consistently scores the lowest, with F1-scores ranging from 0.44 to 0.56. Llama 3.1 scores considerably lower than the encoder-only models, reaching a macro F1 of 0.47 and an accuracy of 0.50. Although Llama 3.1 achieves a high recall of 0.96 for hate speech, its precision for that class is only 0.52, and its F1 for the normal class remains low at 0.35.

On the high agreement subset Table 5, all encoder-only models improve noticeably. BERT and HateBERT reach a macro F1 of 0.79, while ToxicBERT and RoBERTa all score 0.78. Hate speech and normal F1-scores rise to the high 0.80s across all four models. The offensive class remains the weakest class, staying around 0.62 to 0.63. Llama 3.1 improves to a macro F1 of 0.51 and an F1 of 0.42 on the normal class.

On the low agreement subset Table 6, performance drops considerably for all models. The encoder-only models cluster closely together with macro F1-scores between 0.53 and 0.55, and no single model stands out. The normal class is the hardest to classify for three of the four encoder models, BERT F1-score of 0.40, HateBERT and ToxicBERT F1-score at 0.47, whereas for RoBERTa the offensive class is marginally harder to classify than normal (F1 0.48 for offensive versus 0.51 for normal). Llama 3.1 again scores the lowest, with a macro F1 of 0.40 and an accuracy of 0.47. Its F1 for the normal class drops to 0.18, driven by very low recall of 0.10, despite relatively high precision of 0.72.

	BERT			HateBERT			RoBERTa			ToxicBERT			Llama 3.1		
Full Dataset	**Prec.**	**Rec.**	**F1**	**Prec.**	**Rec.**	**F1**	**Prec.**	**Rec.**	**F1**	**Prec.**	**Rec.**	**F1**	**Prec.**	**Rec.**	**F1**
Hate Speech	0.78	0.75	0.76	0.69	0.85	0.76	0.79	0.77	0.78	0.69	0.84	0.76	0.52	0.96	0.68
Offensive	0.55	0.54	0.54	0.59	0.43	0.50	0.56	0.56	0.56	0.60	0.35	0.44	0.35	0.40	0.37
Normal	0.73	0.76	0.74	0.73	0.74	0.74	0.75	0.76	0.75	0.70	0.80	0.75	0.93	0.22	0.35
Accuracy	-	-	0.69	-	-	0.69	-	-	0.71	-	-	0.68	-	-	0.50
Macro avg	0.68	0.68	0.68	0.67	0.67	0.67	0.70	0.70	0.70	0.67	0.66	0.65	0.60	0.53	0.47

Table 4: Full dataset classification report

	BERT			HateBERT			RoBERTa			ToxicBERT			Llama 3.1		
High Dataset	**Prec.**	**Rec.**	**F1**	**Prec.**	**Rec.**	**F1**	**Prec.**	**Rec.**	**F1**	**Prec.**	**Rec.**	**F1**	**Prec.**	**Rec.**	**F1**
Hate Speech	0.87	0.89	0.88	0.83	0.90	0.87	0.81	0.93	0.86	0.85	0.88	0.87	0.60	0.99	0.75
Offensive	0.63	0.62	0.63	0.62	0.65	0.63	0.61	0.63	0.62	0.62	0.62	0.62	0.26	0.52	0.35
Normal	0.87	0.86	0.87	0.89	0.83	0.86	0.90	0.81	0.85	0.87	0.85	0.86	0.99	0.27	0.42
Accuracy	-	-	0.83	-	-	0.82	-	-	0.81	-	-	0.82	-	-	0.53
Macro avg	0.79	0.79	0.79	0.78	0.79	0.79	0.77	0.79	0.78	0.78	0.78	0.78	0.62	0.59	0.51

Table 5: High dataset classification report

	BERT			HateBERT			RoBERTa			ToxicBERT			Llama 3.1		
Low Dataset	**Prec.**	**Rec.**	**F1**	**Prec.**	**Rec.**	**F1**	**Prec.**	**Rec.**	**F1**	**Prec.**	**Rec.**	**F1**	**Prec.**	**Rec.**	**F1**
Hate Speech	0.64	0.65	0.65	0.63	0.70	0.67	0.61	0.68	0.64	0.61	0.70	0.65	0.45	0.91	0.61
Offensive	0.49	0.61	0.55	0.52	0.48	0.50	0.51	0.45	0.48	0.52	0.52	0.52	0.47	0.37	0.41
Normal	0.51	0.33	0.40	0.48	0.47	0.47	0.50	0.52	0.51	0.52	0.43	0.47	0.72	0.10	0.18
Accuracy	-	-	0.55	-	-	0.55	-	-	0.54	-	-	0.56	-	-	0.47
Macro avg	0.55	0.53	0.53	0.54	0.55	0.55	0.54	0.55	0.54	0.55	0.55	0.55	0.55	0.46	0.40

Table 6: Low dataset classification report

4.2 Error Patterns and Misclassifications

The confusion matrices for the full dataset, Figures 2a to 2e, show a consistent pattern across the encoder-only models. Normal content is most often confused with offensive, and offensive content is regularly misclassified as either hate speech or normal. Hate speech and normal content are recognised about equally well; for BERT, normal is even marginally better recognised than hate speech. Llama 3.1 shows a different pattern: true offensive content is mostly predicted as hate speech, but true normal content is mostly predicted as offensive rather than hate speech.

On the high agreement subset, Figures 3a to 3e, the encoder-only models make fewer errors overall. Misclassifications are now concentrated in the offensive class, which is confused with normal somewhat more often than with hate speech. Llama 3.1 continues to over-predict non-normal labels: true offensive content is split roughly evenly between correct predictions and predictions of hate speech, while true normal content is most often predicted as offensive rather than hate speech.

On the low agreement subset, Figures 4a to 4e, misclassifications increase across all models and all classes compared to the high agreement subset. For most encoder-only models, offensive content is misclassified less often here than on the full dataset; RoBERTa is the exception, where offensive content is misclassified more often on this split than on the other two. The normal class remains the weakest class for most models and continues to be confused with offensive more than with hate speech. Llama 3.1 still recognises most hate speech correctly, but it also predicts hate speech for a substantial share of both offensive and normal posts.

4.3 Statistical Significance

In Table 7 you can see the results of McNemar's test for all model pairs across the three dataset splits. On the full dataset, the differences between the four encoder-only models are not statistically significant, except for ToxicBERT versus RoBERTa ($p = 0.0078$). All comparisons between an encoder-only model and Llama 3.1 are highly significant ($p < 0.001$).

On the high agreement subset, no significant differences are found between any of the encoder-only models. All comparisons between an encoder-only model and Llama 3.1 remain highly significant ($p < 0.001$).

On the low agreement subset, the encoder-only models again show no statistically significant differences from one another. All comparisons between an encoder-only model and Llama 3.1 are again highly significant ($p < 0.001$).

Model 1	**Model 2**	**Acc. Model 1**	**Acc. Model 2**	**p-value**
Full Dataset				
BERT	HateBERT	0.689	0.691	0.8261
BERT	ToxicBERT	0.689	0.682	0.3243
BERT	RoBERTa	0.689	0.702	0.0780
BERT	Llama 3.1	0.689	0.502	<0.001***
HateBERT	ToxicBERT	0.691	0.682	0.2564
HateBERT	RoBERTa	0.691	0.702	0.1371
HateBERT	Llama 3.1	0.691	0.502	<0.001***
ToxicBERT	RoBERTa	0.682	0.702	0.0078**
ToxicBERT	Llama 3.1	0.682	0.502	<0.001***
RoBERTa	Llama 3.1	0.702	0.502	<0.001***
High Dataset				
BERT	HateBERT	0.825	0.817	0.3902
BERT	ToxicBERT	0.825	0.818	0.4249
BERT	RoBERTa	0.825	0.812	0.2022
BERT	Llama 3.1	0.825	0.532	<0.001***
HateBERT	ToxicBERT	0.817	0.818	1.0000
HateBERT	RoBERTa	0.817	0.812	0.6108
HateBERT	Llama 3.1	0.817	0.532	<0.001***
ToxicBERT	RoBERTa	0.818	0.812	0.6018
ToxicBERT	Llama 3.1	0.818	0.532	<0.001***
RoBERTa	Llama 3.1	0.812	0.532	<0.001***
Low Dataset				
BERT	HateBERT	0.546	0.554	0.5756
BERT	ToxicBERT	0.546	0.555	0.4610
BERT	RoBERTa	0.546	0.544	0.8842
BERT	Llama 3.1	0.546	0.469	<0.001***
HateBERT	ToxicBERT	0.554	0.555	1.0000
HateBERT	RoBERTa	0.554	0.544	0.3784
HateBERT	Llama 3.1	0.554	0.469	<0.001***
ToxicBERT	RoBERTa	0.555	0.544	0.3971
ToxicBERT	Llama 3.1	0.555	0.469	<0.001***
RoBERTa	Llama 3.1	0.544	0.469	<0.001***

* p<0.05 ** p<0.01 *** p<0.001

Table 7: McNemar's test results across all dataset splits

4.4 False Positive rate on True-Normal content

Table 8 reports the false positive rate on true-normal content, broken down by target community, for all five models across the three dataset splits, together with the number of test instances underlying each group.

On the Full split, the false positive rate for content with no identifiable target ("None") ranges from 0.170 to 0.181 for the four encoder-only models. For every target-community category (race, gender, religion, sexuality, and other), the false positive rate is higher than this for all four models, ranging from 0.253 to 0.469 across these categories. The same pattern holds on the Low split: the false positive rate for "None" ranges from 0.387 to 0.516, while every target-community category ranges from 0.438 to 0.877, again higher than "None" for all four models.

On the High split, this pattern is less uniform across categories and models. Gender, other, and sexuality show a higher false positive rate than "None" for all four encoder-only models. For race, this holds for BERT, HateBERT, and ToxicBERT, but not for RoBERTa, where race (0.141) is lower than "None" (0.173). For religion, the false positive rate is lower than "None" for BERT and ToxicBERT, higher for HateBERT and RoBERTa, and equal in magnitude for RoBERTa relative to "None" (0.179 versus 0.173).

Llama 3.1's false positive rate on normal content exceeds that of every encoder-only model, for every target group and on every split, ranging from 0.529 to 1.000. Across the target-community categories within a single split, its false positive rate varies less than that of the encoder-only models: as can be seen on the Full split, it ranges from 0.750 to 0.888 across race, gender, religion, and sexuality, compared to a wider range of 0.253 to 0.469 for the encoder-only models on the same categories.

The number of test instances underlying each group varies considerably. The disability category comprises six instances on the Full split, four on the Low split, and no instances on the High split. The other categories range from 34 instances (sexuality, High and Low splits) to 576 instances ("None", High split).

Model	Disability	Gender	None	Other	Race	Religion	Sexuality
Full Dataset							
BERT	0.833	0.345	0.170	0.306	0.313	0.405	0.302
HateBERT	1.000	0.405	0.181	0.316	0.300	0.469	0.365
RoBERTa	0.667	0.379	0.177	0.276	0.253	0.451	0.427
ToxicBERT	1.000	0.371	0.170	0.337	0.352	0.387	0.365
Llama 3.1	1.000	0.888	0.736	0.867	0.816	0.766	0.750
n	*6*	*116*	*576*	*98*	*233*	*111*	*96*
High Dataset							
BERT	–	0.236	0.099	0.250	0.165	0.077	0.206
HateBERT	–	0.292	0.137	0.295	0.188	0.179	0.235
RoBERTa	–	0.333	0.173	0.250	0.141	0.179	0.206
ToxicBERT	–	0.250	0.124	0.205	0.200	0.077	0.235
Llama 3.1	–	0.792	0.732	0.841	0.788	0.564	0.529
n	*0*	*72*	*533*	*44*	*85*	*39*	*34*
Low Dataset							
BERT	1.000	0.479	0.516	0.673	0.647	0.877	0.618
HateBERT	0.500	0.500	0.484	0.539	0.559	0.778	0.588
RoBERTa	0.750	0.438	0.387	0.635	0.559	0.704	0.471
ToxicBERT	0.750	0.438	0.484	0.577	0.529	0.790	0.441
Llama 3.1	1.000	0.917	0.903	0.904	0.918	0.877	0.824
n	*4*	*48*	*31*	*52*	*170*	*81*	*34*

Table 8: False positive rate on true-normal content, broken down by target community.

4.5 False Negative Rate on True-Hate-Speech Content

Table 9 reports the false negative rate on true-hate-speech content, broken down by target community, for all five models across the three dataset splits, together with the number of test instances underlying each group.

On the Full split, the false negative rate for the four encoder-only models is lowest for race, ranging from 0.135 to 0.177, and for religion, ranging from 0.170 to 0.220, and highest for gender, ranging from 0.517 to 0.586, and sexuality, ranging from 0.451 to 0.476. The same ordering appears on the Low split, where race, ranging from 0.192 to 0.246, and religion, ranging from 0.266 to 0.308, again have the lowest false negative rate, and gender, ranging from 0.375 to 0.625, and sexuality, ranging from 0.472 to 0.698, the highest. On the High split, race, ranging from 0.034 to 0.084, and religion, ranging from 0.047 to 0.147, remain the lowest for all four encoder-only models, while sexuality, ranging from 0.344 to 0.469, is the highest category on this split.

Llama 3.1's false negative rate on hate speech is lower than that of every encoder-only model in every target group and on every split, ranging from 0.000 to 1.000 depending on the group. For race and religion specifically, its false negative rate stays below 0.08 on all three splits.

The disability category comprises three instances on the Full split and one on the Low split, and no instances on the High split. The none category, meaning hate speech content with no identified target, comprises one instance on the Full split and two on the Low split, and no instances on the High split. the High split.

Model	Disability	Gender	None	Other	Race	Religion	Sexuality
Full Dataset							
BERT	0.333	0.552	1.000	0.542	0.160	0.214	0.463
HateBERT	0.333	0.517	1.000	0.542	0.152	0.199	0.451
RoBERTa	0.667	0.517	1.000	0.500	0.135	0.170	0.463
ToxicBERT	0.667	0.586	1.000	0.625	0.177	0.220	0.476
Llama 3.1	0.333	0.241	1.000	0.250	0.019	0.024	0.110
n	*3*	*29*	*1*	*24*	*481*	*336*	*82*
High Dataset							
BERT	–	0.000	–	0.500	0.067	0.147	0.344
HateBERT	–	0.111	–	0.500	0.059	0.089	0.406
RoBERTa	–	0.222	–	0.333	0.034	0.047	0.406
ToxicBERT	–	0.000	–	0.333	0.084	0.094	0.469
Llama 3.1	–	0.000	–	0.000	0.008	0.005	0.063
n	*0*	*9*	*0*	*6*	*238*	*191*	*32*
Low Dataset							
BERT	1.000	0.625	1.000	0.524	0.246	0.308	0.698
HateBERT	1.000	0.500	1.000	0.524	0.192	0.266	0.472
RoBERTa	1.000	0.375	1.000	0.524	0.241	0.284	0.491
ToxicBERT	1.000	0.438	1.000	0.571	0.205	0.272	0.585
Llama 3.1	1.000	0.313	1.000	0.286	0.076	0.047	0.075
n	*1*	*16*	*2*	*21*	*224*	*169*	*53*

Table 9: False negative rate on true-hate-speech content, broken down by target community.

4.6 Qualitative Examples of Misclassification

While Sections 4.2 to 4.5 quantify how often each model fails and on which classes, the examples below illustrate what these failures look like in practice. Each example lists the true label, the target community, and the label predicted by each of the five models.

> ***Content warning:*** *the example below is an authentic post from the HateXplain dataset (sourced from Gab and Twitter) and contains an explicit racial slur alongside graphic, degrading language. It is included unedited because the offensive language itself is what the model must learn to recognise; readers who would prefer to skip it can continue at Section 5.*

A. Llama 3.1 over-predicting hate speech on offensive content

Example A1 (target: race, true label: offensive) — **BERT**: offensive, **HateBERT**: normal, **ToxicBERT**: offensive, **RoBERTa**: normal, **Llama**: hate speech.

> `"why do white countries pay so much foreign aid and also give refugees money in our own countries why the double dip one or the other or better yet none at all enough is enough we are backing dead horses"`

Example A2 (target: sexuality, true label: offensive) — **BERT**: offensive, **HateBERT**: offensive, **ToxicBERT**: offensive, **RoBERTa**: offensive, **Llama**: hate speech.

> `"wea df all you saints fans come from [emoji] go back to yo faggot ass cowbitches saints won [emoji] like bitch who are you [emoji]"`

B. Encoder models missing hate speech targeting gender and sexuality

Section 4.5 shows that hate speech concerning gender and sexuality is missed by encoder models more often than hate speech concerning race or religion. The following examples show this failure directly.

Example B1 (target: sexuality, true label: hate speech) — **BERT**: normal, **HateBERT**: normal, **ToxicBERT**: normal, **RoBERTa**: offensive, **Llama**: hate speech.

> `"<user> anyone i see with a pixie cut or a half shaved thing at work i call dykes they are always the customers bitching at work and wanting to speak to a manager whenever we see one it here comes a karen jajajaja [emoji] the ones con bad multi color hair of"`

Example B2 (target: gender, true label: hate speech) — **BERT**: normal, **HateBERT**: normal, **ToxicBERT**: normal, **RoBERTa**: normal, **Llama**: hate speech.

> `"<user> when it comes to the issue of women facing domestic violence i will beat them like a drum"`

In Example B2 in particular, all four fine-tuned encoder models miss a post that expresses direct intent to harm a group defined by gender, while only the zero-shot model recognises it. This is a clear instance of the false-negative risk discussed in Section 5.3: harmful content that a system, whose purpose is to protect people, allows to remain visible.

C. False positives on normal content concerning race and religion

Section 4.4 shows that normal content concerning race and religion is flagged as harmful considerably more often than content with no identifiable target. The examples below show what such a false positive looks like in practice.

Example C1 (target: race, true label: normal) — **BERT**: offensive, **HateBERT**: offensive, **ToxicBERT**: normal, **RoBERTa**: hate speech, **Llama**: hate speech.

```
"giving the asians a bad name"
```

Example C2 (target: religion, true label: normal) — **BERT**: offensive, **HateBERT**: hate speech, **ToxicBERT**: offensive, **RoBERTa**: offensive, **Llama**: hate speech.

```
"when jews call out jews you know its bad"
```

Neither post expresses hatred or targets a group; both simply reference a group by name in an otherwise unremarkable statement. Yet most models flag them as harmful, illustrating the silencing risk discussed in Section 5.4: posts that merely mention a racial or religious group appear to trigger a harmful label independent of actual content.

D. The offensive class as a boundary case

Section 5.2 argues that the offensive class is difficult to classify because it sits at the boundary between hate speech and normal content. The examples below show models disagreeing with each other on exactly this boundary for the same post.

Example D1 (target: race, true label: offensive) — **BERT**: hate speech, **HateBERT**: hate speech, **ToxicBERT**: normal, **RoBERTa**: hate speech, **Llama**: hate speech.

```
"protip when a nigger says mah dude you are not his dude"
```

Example D2 (target: race, true label: offensive) — **BERT**: normal, **HateBERT**: normal, **ToxicBERT**: normal, **RoBERTa**: hate speech, **Llama**: hate speech.

```
"<user> od leave them white devils alone"
```

In both cases, models disagree not on whether the post is harmless, but on how harmful it is — some models escalate the same post to hate speech that others consider unremarkable, showing that the offensive/hate speech boundary is unstable even within a single model type.

4.7 Training Convergence

To verify that fine-tuning proceeded correctly, the training and evaluation loss were tracked at each logged step for all four encoder-only models across all three dataset splits. Figure 1 shows these curves, a summary table of the loss at the end of each epoch is given in Table 10.

Across all twelve training runs, the training loss decreases steadily and without instability, indicating that each model was learning consistently from the training data. The evaluation loss, however, follows a different pattern: it decreases from epoch 1 to epoch 2, then rises sharply in

epoch 3, while the training loss continues to fall. This divergence between training and evaluation loss is a sign of overfitting: by the third epoch, each model increasingly memorises specific details of the training set rather than learning patterns that generalise.

This is not a failure of the training setup. Because `load_best_model_at_end=True` was used (see Table 2), the Trainer automatically retained the checkpoint with the lowest evaluation loss for each run — in every case, the checkpoint from epoch 2, before the onset of overfitting — rather than the final, more overfit checkpoint from epoch 3. The results reported in Sections 4.1 to 4.6 are therefore based on the best-performing checkpoint per run, not the most overfit one. This finding also foreshadows a limitation discussed later in Section 5.5: because checkpoint selection used the same test set later used for final evaluation, the reported test performance is likely a slight overestimate of true generalisation, since the checkpoint itself was chosen based on performance on that exact data.

Model	Split	epoch 1	epoch 2	epoch 3	Final eval loss
BERT	Full	0.6786	0.5944	0.3849	0.8994
BERT	High agreement	0.5627	0.3113	0.3113	0.6053
BERT	Low agreement	0.9509	0.7091	0.7091	1.0829
HateBERT	Full	0.6357	0.5467	0.3269	0.9604
HateBERT	High agreement	0.5177	0.2540	0.2540	0.6552
HateBERT	Low agreement	0.9091	0.6669	0.6669	1.0814
RoBERTa	Full	0.6735	0.5913	0.4058	0.8696
RoBERTa	High agreement	0.5300	0.2928	0.2928	0.6538
RoBERTa	Low agreement	0.9326	0.7084	0.7084	1.1195
ToxicBERT	Full	0.6688	0.5748	0.3647	0.9561
ToxicBERT	High agreement	0.5379	0.2924	0.2924	0.5964
ToxicBERT	Low agreement	0.9044	0.6273	0.6273	1.1781

Table 10: Training loss at the end of each epoch, and the final evaluation loss of the checkpoint selected via `load_best_model_at_end=True`, for all four encoder-only models across the three dataset splits.

For the Full split, training loss was logged at five points, giving a distinct value for each of the three epochs. For the High and Low agreement splits, however, the smaller training sets meant training loss was logged at only two points, which happened to coincide with the boundary between epoch 2 and epoch 3; the training loss reported for these two epochs in Table 10 is therefore identical. This is an artefact of the logging frequency relative to the size of these smaller splits, not an indication that training stalled between these two epochs — the evaluation loss, logged separately once per epoch, continues to change as expected across all three epochs.

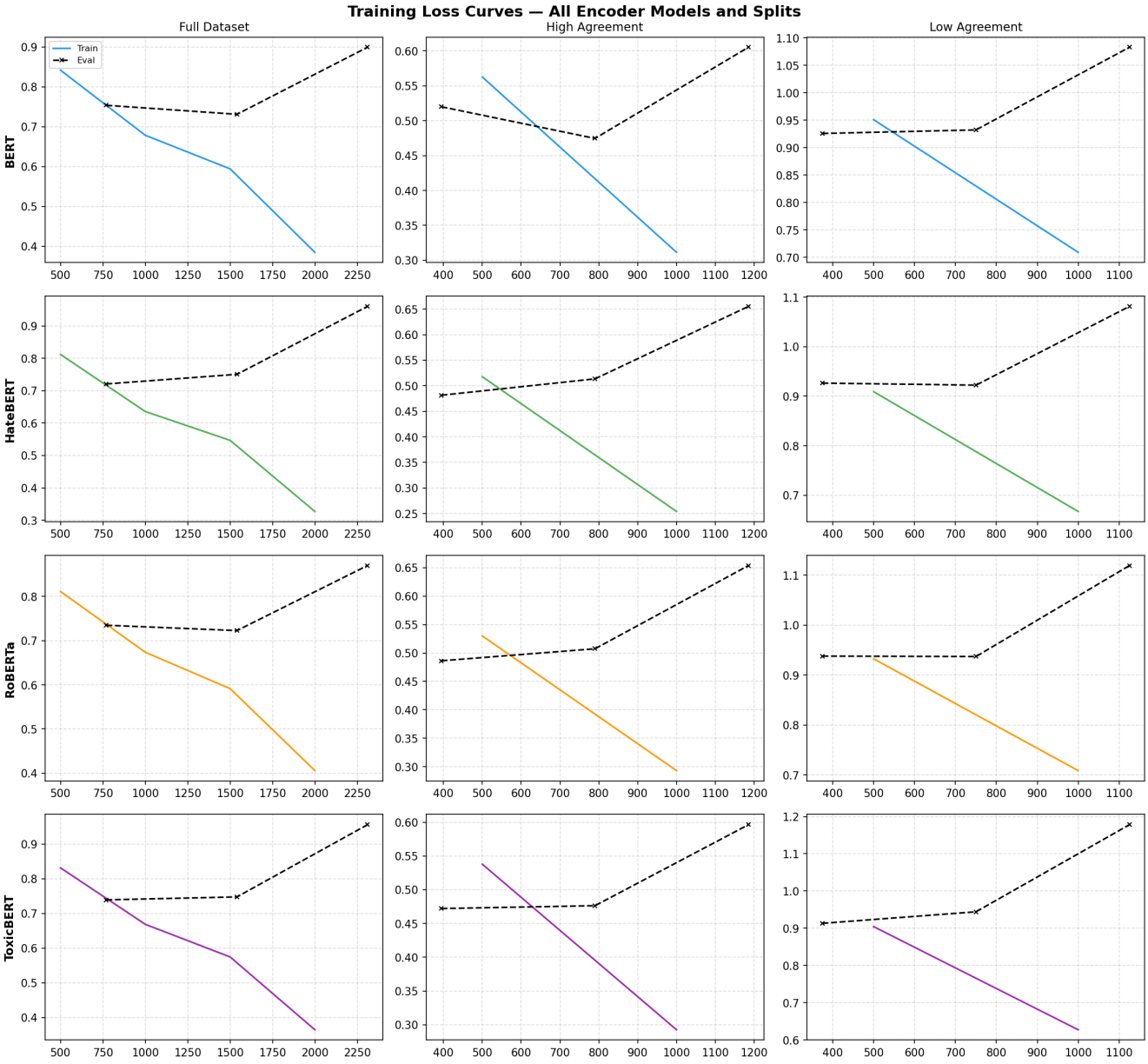


Figure 1: Training and evaluation loss curves for all four encoder-only models BERT, HateBERT, RoBERTa and ToxicBERT across the three dataset splits, Full, High Agreement and Low Agreement. The solid coloured line shows the training loss at each logged step; the black dashed line shows the evaluation loss recorded at the end of each epoch.

5 Discussion

This section discusses the results presented in Section 4 and considers what they reveal about automated hate speech detection more broadly. Section 5.1 discusses how annotator agreement shapes model performance and how this affects the comparison between the encoder-only models and Llama 3.1. Section 5.2 examines why the offensive class is consistently the hardest to classify. Section 5.3 considers the different ways in which Llama 3.1 and the encoder models fail, and the ethical risks that follow from each. Section 5.4 extends this discussion to whether these errors disproportionately affect content concerning marginalised communities. The section closes with the limitations of this study in Section 5.5.

5.1 Ambiguity and Model Performance

The findings of this study confirm what earlier research has already hinted at: model performance in hate speech detection is not just a matter of architecture or training quality, but is fundamentally shaped by the nature of the data itself. When the content is clear enough for humans to agree on, models handle it well. When humans themselves are unsure, models are too. This pattern held consistently across all five models and all three dataset splits. It makes it difficult to attribute to any single architecture or training approach. It instead points to something more fundamental about the task itself: hate speech does not have clean boundaries, and that ambiguity is reflected directly in how models perform. No model, however well trained, will fully escape that.

A second pattern concerns the difference between the fine-tuned encoder models and the zero-shot decoder model. The encoder models consistently and significantly outperformed Llama 3.1 across all dataset splits. What is more telling is that this advantage shrinks when the content becomes more ambiguous. When annotators were divided, the gap was smaller. Fine-tuning on majority labels gives encoder models a clear advantage on content where human agreement is high, but that advantage fades when the content sits in the grey area between categories. As discussed in Section 2.6, majority labels carry less reliable signal on ambiguous instances, and models trained on them will reflect that. Llama 3.1 does not have that same dependency on majority consensus, which may partly explain why the gap narrows on harder content.

5.2 The Difficulty of the Offensive Class

A pattern that stands out across most models and dataset splits is the consistent difficulty with the offensive class. On the full and high agreement subsets, every encoder-only model struggled most with offensive content, while hate speech and normal content were both comparatively well recognised. On the low agreement subset, offensive remains the hardest class for three of the four encoder models, with RoBERTa as a marginal exception, where normal is instead the hardest class by a small margin. This is not simply a matter of model weakness. It reflects something inherent about the category itself. Offensive content sits at the boundary between hate speech and normal content: it is language that is rude or aggressive but does not rise to the level of targeting a group, a distinction that is inherently harder to draw than the line between clearly hateful and clearly harmless content. That is precisely the boundary where human annotators tend to disagree most, as content that one person reads as merely offensive, another may read as either hate speech or harmless. That disagreement carries through into the labels that models learn from, meaning

models are trying to learn a boundary that was never clearly defined in the first place. This points to a broader limitation of framing hate speech detection as a clean three-way classification problem, when in reality the boundaries between categories are fluid and heavily dependent on context, background, and personal interpretation.

5.3 Divergent Error Patterns and Ethical Risks

A particularly important finding concerns the error pattern of Llama 3.1. Across all dataset splits, Llama 3.1 showed a strong tendency to over-predict harmful labels, though the specific label it defaults to differs depending on the true class. True offensive content is mostly classified as hate speech, whereas true normal content is mostly classified as offensive rather than hate speech. This kind of systematic over-prediction is ethically significant regardless of which harmful label is assigned. It means that a large share of the errors made by this model are false positives: content that is not hateful, and in many cases not even offensive, being flagged as if it were. In the context of automated content moderation, false positives are not a neutral mistake. They result in the silencing of voices that should not be silenced, a risk discussed in more detail in Section 2.5. A model deployed to protect people from harmful content can, through systematic over-prediction, end up restricting the very groups it was intended to protect.

While Llama 3.1 tends to over-predict harmful labels, the encoder models show the opposite problem on ambiguous content. Where fine-tuned models miss hateful content that is subtle or hard to categorise, that content remains visible and its targets are left without protection. The most ambiguous cases are not necessarily the least harmful ones. Harmful intent does not disappear simply because it is hard to pin down. What the results reveal is that the two types of models fail in fundamentally different directions. One tends to over-flag, risking the silencing of legitimate speech. The other tends to under-flag on hate speech specifically, risking that harmful content goes unchecked. Both failures carry ethical weight, and both are most pronounced on exactly the content that is hardest to judge. When human judgment is uncertain, model errors are not just technical shortcomings but have real consequences for real people.

5.4 Disproportionate Impact on Marginalised Communities

The false positive rate on normal content, broken down by target community (Table 8), adds an important dimension to this picture of silencing. For race and religion specifically, every encoder-only model misclassified normal content concerning these groups considerably more often than normal content with no identifiable target, on both the full and low agreement subsets. This is not a marginal effect: the false positive rate for these categories was between roughly one and a half and nearly three times higher than for content with no target. This supports the concern raised in Section 2.5, and in prior work such as Sap et al. (2019) [SCG+19], that automated hate speech detection does not fail evenly across the population it is meant to protect.

The false negative rate on hate speech, broken down by target community Table 9, completes this picture of disproportionate impact by addressing the other side of RQ3. Here the pattern runs in the opposite direction to the false positive rate: for every encoder-only model, hate speech concerning race or religion is missed less often than hate speech concerning gender or sexuality, and this ordering holds on all three dataset splits. Race and religion are therefore the target communities for which encoder-only models are least likely to leave hate speech undetected, yet

at the same time they are the target communities for which those same models are most likely to wrongly flag normal content. Gender and sexuality show the reverse pattern: encoder-only models miss hate speech concerning these groups more often, while normal content concerning these groups is flagged somewhat less disproportionately than content concerning race or religion. No single target community in this study is therefore subject to both forms of harm at once from the encoder-only models; instead, the two ethical risks identified in Section 5.3, silencing through false positives and under-protection through false negatives, fall unevenly across different communities. Llama 3.1 misses hate speech less often than every encoder-only model across all target groups, which is consistent with its general tendency to over-predict harmful labels rather than with a bias that specifically protects any one community.

5.5 Limitations

One limitation of this study is the way Llama 3.1 and the encoder models were compared. The encoder models were trained on labeled examples from the same task, while Llama 3.1 was given no training at all and had to rely on its general knowledge. The difference in performance between the two cannot simply be explained by how they are built. A version of Llama 3.1 that was trained on the same data might perform very differently, but doing that was not possible within the scope of this study due to the computing power it would require.

The correct labels used to evaluate all models in this study were decided by majority vote across three people. On the low agreement subset, this means two out of three annotators determined what the right answer was. That label is not an objective truth. It is a practical choice, and models are judged against it. This makes it harder to say with confidence what it really means for a model to be right or wrong on content that even humans found hard to judge.

All data in this study came from Twitter and Gab. These are specific platforms with their own cultures and types of content. Whether the findings hold for other platforms, other languages, or other online spaces is something this study cannot say. Hate speech takes different forms in different places, and results that hold here may not hold elsewhere.

Running Llama 3.1 70B required a dedicated server at LIACS and a specific local setup that is not easy to replicate. This means other researchers cannot straightforwardly reproduce the Llama results in the same way they could for the encoder models, which run on standard academic computing resources.

A further limitation concerns the number of statistical tests conducted. McNemar's test was run on every pair of models within each dataset split, resulting in 30 separate tests in total (ten model pairs across each of the three splits). No correction for multiple comparisons was applied. Running this many tests increases the chance that at least one result appears statistically significant purely by chance, even when no real difference exists. The one borderline result found among the four encoder-only models, ToxicBERT versus RoBERTa on the full dataset ($p = 0.0078$), is a plausible candidate for such a false alarm.

Another limitation concerns the way the best checkpoint was selected during fine-tuning. For all four encoder-only models, `evaluation_strategy="epoch"` together with `load_best_model_at_end=True` meant that the Trainer selected the checkpoint with the lowest evaluation loss on the same test set that was later used to report the classification results in Section 4. This is a form of data leakage: the checkpoint itself was chosen based on its performance on the exact data used for the final evaluation, rather than on a held-out validation set that plays no role in the reported metrics. In

practice, this leakage is limited in scope, since the choice was between at most three checkpoints per model per split rather than an extensive hyperparameter search, but it means the reported test performance is likely a slight overestimate of how these models would perform on genuinely unseen data. A more rigorous setup would use a three-way split, with a separate validation set for checkpoint selection and a test set reserved exclusively for final evaluation; this was not done here due to the additional computational cost of retraining all models under a new split.

6 Conclusion

In response to RQ1, all models performed better on the high agreement subset than on the low agreement subset. RoBERTa performed best among the encoder models, but differences between the four encoder models were small and mostly not statistically significant. Llama 3.1 scored the lowest across all splits. The offensive class was the hardest to classify for most models, though for RoBERTa the normal class was marginally harder on the low agreement subset.

In response to RQ2, the encoder models outperformed Llama 3.1 across all splits, but the gap was smaller on the low agreement subset. Fine-tuning on majority labels gives encoder models an advantage on clear content, but that advantage fades when content is ambiguous. Llama 3.1 responds to ambiguous content differently, but not better overall.

In response to RQ3, the two model types failed in opposite directions. Llama 3.1 over-predicted harmful labels, producing many false positives, most often predicting hate speech for offensive content and offensive for normal content. The encoder models missed more hate speech on the low agreement subset, producing more false negatives. Both failures were most pronounced on the hardest content. For the encoder models, the risk of false positives on normal content was more pronounced for race and religion, while the risk of false negatives on hate speech was more pronounced for gender and sexuality: no single target community was subject to both risks at once.

Taken together, these findings address the main research question: model errors are directly tied to annotator disagreement. Where humans could not agree, models struggled most. This holds across all five models and all splits. Automated hate speech detection is least reliable exactly where it matters most.

Future research could look at training models with soft labels instead of majority labels. Rather than forcing a single label onto content that annotators disagreed on, soft labels keep that uncertainty in the training data. This could help models handle ambiguous content better, since they would no longer be trained as if every case has a clear correct answer.

References

[AID+25] Aish Albladi, Minarul Islam, Amit Das, Maryam Bigonah, Zheng Zhang, Fatemeh Jamshidi, Mostafa Rahgouy, Nilanjana Raychawdhary, Daniela Marghitu, and Cheryl Seals. Hate speech detection using large language models: A comprehensive review. *IEEE Access*, 13:20871–20892, 2025.

[Bro17] Alexander Brown. What is hate speech? part 1: The myth of hate. *Law and philosophy*, 36(4):419–468, 2017.

[CAG25] Mahima Choudhary, Basant Agarwal, and Vishnu Goyal. Hate speech detection: leveraging llm-gpt2 with fine-tuning and multi-shot techniques. *Procedia Computer Science*, 258:2817–2825, 2025.

[CBMG21] Tommaso Caselli, Valerio Basile, Jelena Mitrović, and Michael Granitzer. Hatebert: Retraining bert for abusive language detection in english. In *Proceedings of the 5th Workshop on Online Abuse and Harms (WOAH 2021)*, pages 17–25, 2021.

[DDP22] Aida Mostafazadeh Davani, Mark Díaz, and Vinodkumar Prabhakaran. Dealing with disagreements: Looking beyond the majority vote in subjective annotations. *Transactions of the Association for Computational Linguistics*, 10:92–110, 2022.

[DGPGPC18] Ona De Gibert, Naiara Perez, Aitor García-Pablos, and Montse Cuadros. Hate speech dataset from a white supremacy forum. In *Proceedings of the 2nd workshop on abusive language online (ALW2)*, pages 11–20, 2018.

[DSY25] Somaiyeh Dehghan, Mehmet Umut Sen, and Berrin Yanikoglu. Dealing with annotator disagreement in hate speech classification. *arXiv preprint arXiv:2502.08266*, 2025.

[DWMW17] Thomas Davidson, Dana Warmsley, Michael Macy, and Ingmar Weber. Automated hate speech detection and the problem of offensive language. In *Proceedings of the international AAAI conference on web and social media*, volume 11, pages 512–515, 2017.

[Els22] Fatma Elsafoury. Darkness can not drive out darkness: Investigating bias in hate speechdetection models. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop*, pages 31–43, 2022.

[FBP+20] Komal Florio, Valerio Basile, Marco Polignano, Pierpaolo Basile, and Viviana Patti. Time of your hate: The challenge of time in hate speech detection on social media. *Applied Sciences*, 10(12):4180, 2020.

[FDWT22] Paula Fortuna, Monica Dominguez, Leo Wanner, and Zeerak Talat. Directions for nlp practices applied to online hate speech detection. In *Proceedings of the 2022 conference on empirical methods in natural language processing*, pages 11794–11805, 2022.

[FN18] Paula Fortuna and Sérgio Nunes. A survey on automatic detection of hate speech in text. *Acm Computing Surveys (Csur)*, 51(4):1–30, 2018.

[GDF25] Faeze Ghorbanpour, Daryna Dementieva, and Alexandar Fraser. Can prompting llms unlock hate speech detection across languages? a zero-shot and few-shot study. In *Proceedings of the The 9th Workshop on Online Abuse and Harms (WOAH)*, pages 413–425, 2025.

[GDJ+24] Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

[HT20] Laura Hanu and Unitary Team. Detoxify. *Zenodo*, 2020.

[Kor21] Mikhail V Koroteev. Bert: a review of applications in natural language processing and understanding. *arXiv preprint arXiv:2103.11943*, 2021.

[LOG+19] 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. *arXiv preprint arXiv:1907.11692*, 2019.

[MAA26] Joshua Muhumuza, Joab Ezra Agaba, and Mercy Amiyo. Majority vote silences minority values: Annotator disagreement at the hate/offensive boundary in hatexplain. *arXiv preprint arXiv:2606.28772*, 2026.

[McN47] Quinn McNemar. Note on the sampling error of the difference between correlated proportions or percentages. *Psychometrika*, 12(2):153–157, 1947.

[MSY+21] Binny Mathew, Punyajoy Saha, Seid Muhie Yimam, Chris Biemann, Pawan Goyal, and Animesh Mukherjee. Hatexplain: A benchmark dataset for explainable hate speech detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 14867–14875, 2021.

[MYY+19] Sean MacAvaney, Hao-Ren Yao, Eugene Yang, Katina Russell, Nazli Goharian, and Ophir Frieder. Hate speech detection: Challenges and solutions. *PloS one*, 14(8):e0221152, 2019.

[MZ21] Nanlir Sallau Mullah and Wan Mohd Nazmee Wan Zainon. Advances in machine learning algorithms for hate speech detection in social media: a review. *IEEE access*, 9:88364–88376, 2021.

[SCG+19] Maarten Sap, Dallas Card, Saadia Gabriel, Yejin Choi, and Noah A Smith. The risk of racial bias in hate speech detection. In *Proceedings of the 57th annual meeting of the association for computational linguistics*, pages 1668–1678, 2019.

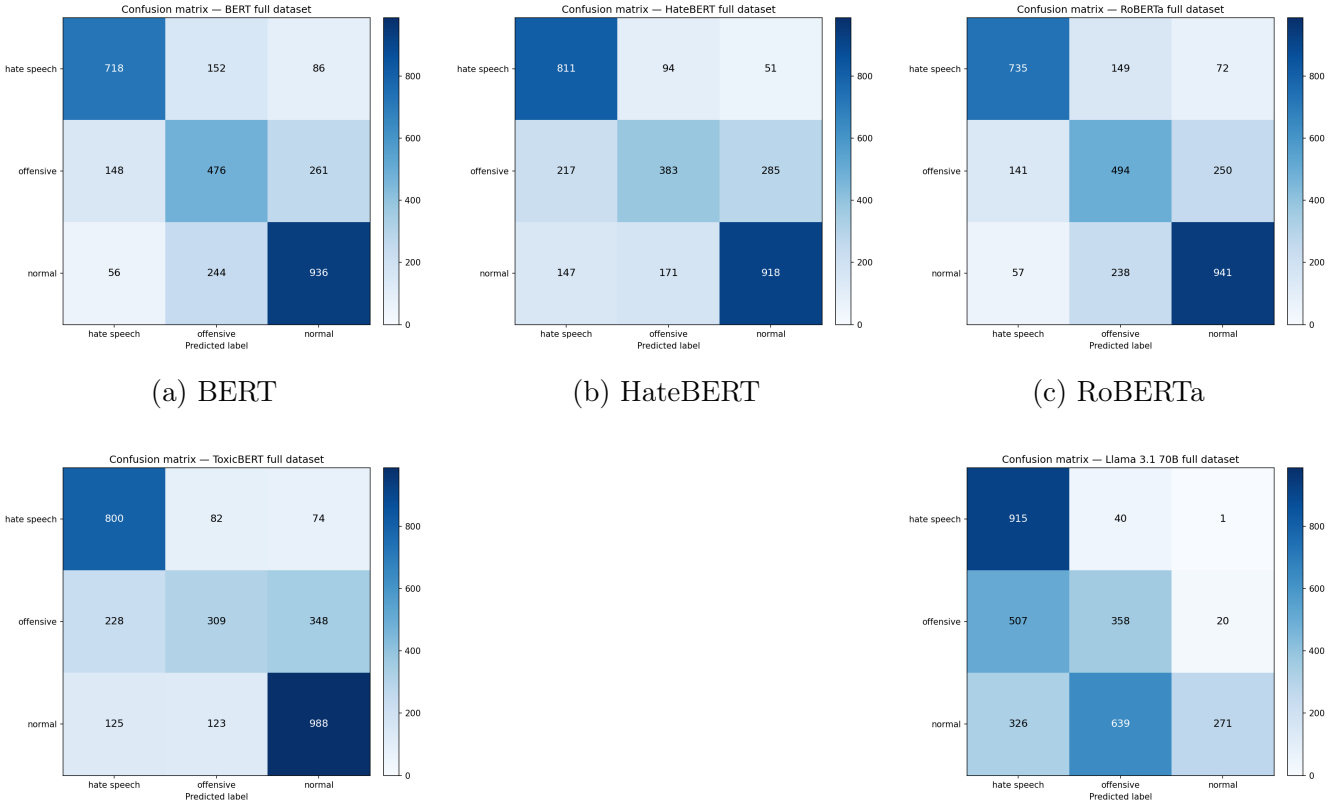
[VCH+23] Francielle Vargas, Isabelle Carvalho, Ali Hürriyetoğlu, Thiago Pardo, and Fabrício Benevenuto. Socially responsible hate speech detection: Can classifiers reflect social stereotypes? In *Proceedings of the 14th international conference on recent advances in natural language processing*, pages 1187–1196, 2023.

[YZ21] Wenjie Yin and Arkaitz Zubiaga. Towards generalisable hate speech detection: a review on obstacles and solutions. *PeerJ computer science*, 7:e598, 2021.

[ZZL+23] Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, et al. A survey of large language models. *arXiv preprint arXiv:2303.18223*, 1(2):1–124, 2023.

A Confusion Matrices

A.1 Full Dataset

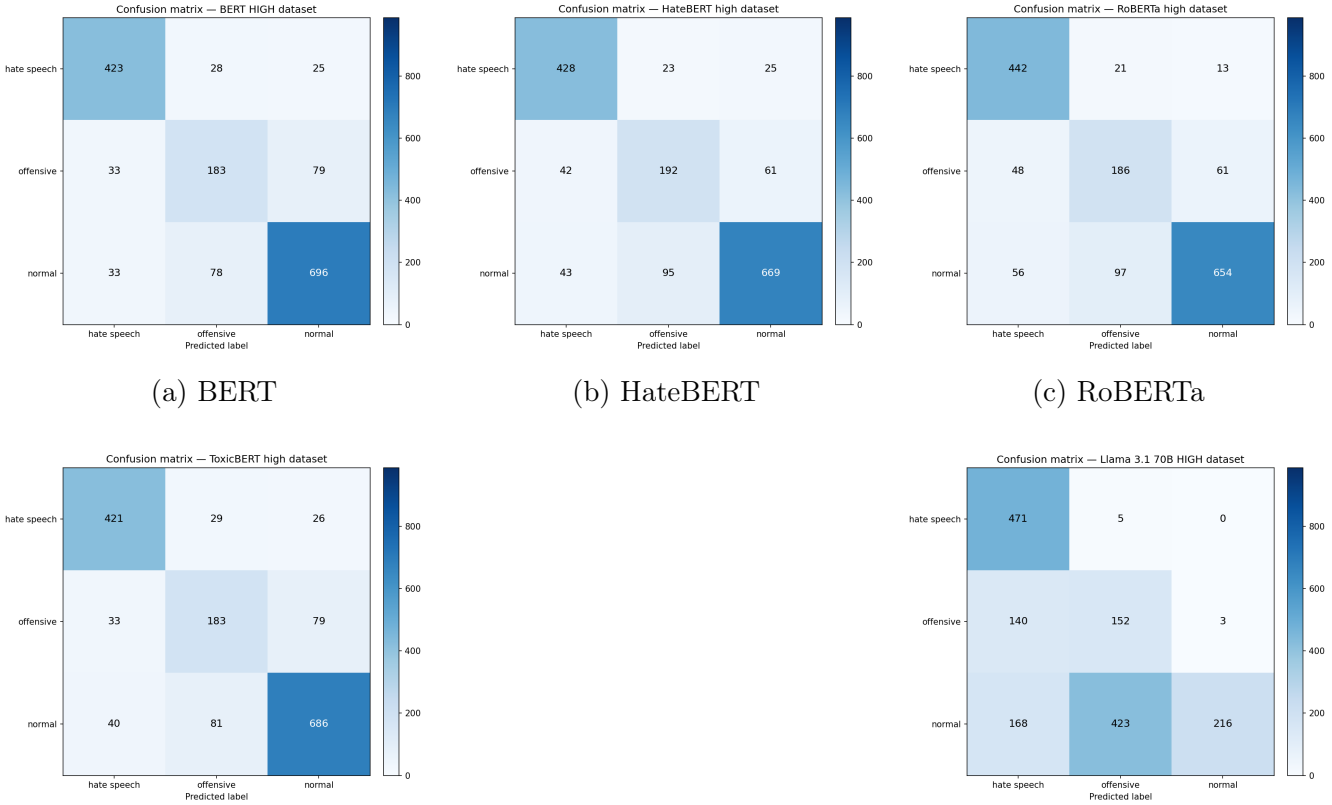


(a) BERT (b) HateBERT (c) RoBERTa

(d) ToxicBERT (e) Llama 3.1

Figure 2: Confusion matrices — Full dataset

A.2 High Dataset

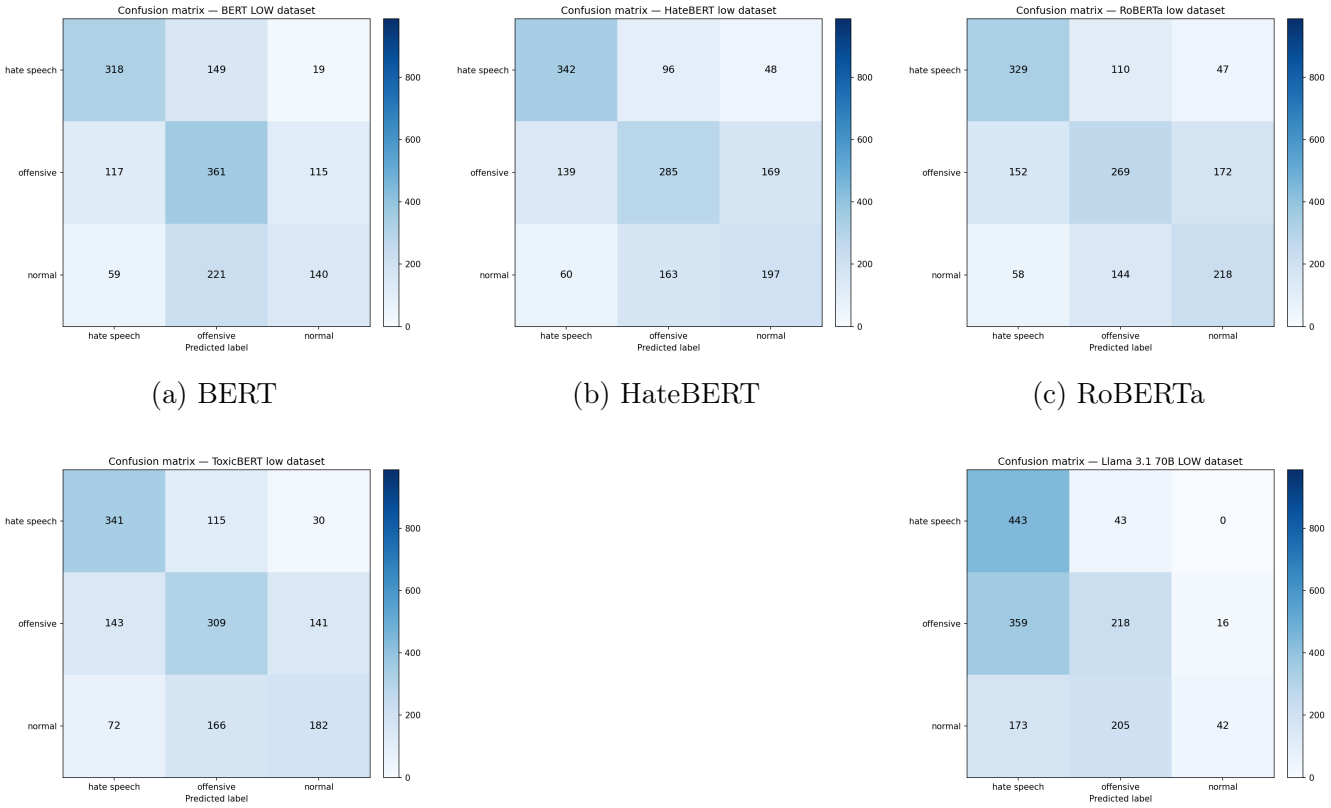


(a) BERT (b) HateBERT (c) RoBERTa

(d) ToxicBERT (e) Llama 3.1

Figure 3: Confusion matrices — High agreement dataset

A.3 Low Dataset



(a) BERT (b) HateBERT (c) RoBERTa

(d) ToxicBERT (e) Llama 3.1

Figure 4: Confusion matrices — Low agreement dataset