VRIJE UNIVERSITEIT AMSTERDAM

MASTER'S THESIS

Modeling Offensive Language as a Distinct Class for Hate Speech Detection

Author:
Areumbyeol KIM

Supervisors:
prof. dr. Antske FOKKENS
dr. H. D. VAN DER VLIET

Second reader:
E. MAKS

A thesis submitted in fulfillment of the requirements for the degree of MA Linguistics

in the

Computational Linguistics and Text Mining Lab
Faculty of Humanities

August 18, 2025

Declaration of Authorship

I, Areumbyeol KIM, declare that this thesis titled, Modeling Offensive Language as a Distinct Class for Hate Speech Detection" and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed: Areumbyeol Kim
Date:15-08-2025

Content note

This thesis includes examples of offensive language and hate speech, presented solely for research purposes. Reader discretion is advised. Where the exact form is not essential, terms are partially masked, otherwise they are reproduced verbatim.

"If liberty means anything at all,
it means the right to tell people what they do not want to hear."

— George Orwell

VRIJE UNIVERSITEIT AMSTERDAM

Abstract

Faculty of Humanities
Department of Language and Communication

MA Linguistics

Modeling Offensive Language as a Distinct Class for Hate Speech Detection

by Areumbyeol KIM

This thesis explores how modeling offensive but not hateful language as a distinct class impacts the task of detection of hate speech. Using a ternary classification scheme (Hateful, Offensive, Clean), I evaluate a RoBERTa-base model in the full three-class setup and in binary variants where two classes are merged or the offensive class is removed (*Hate vs. Non-hate*, *Non-clean vs. Clean*, and *Hate vs. Clean*). To probe model behavior beyond set-internal performance, I revise both HateCheck—a fine-grained diagnostic suite—and an existing extension, aligning them with the ternary system by re-annotating them and correcting errors. Set-internally, the main challenge is distinguishing between hateful and offensive content, whereas "clean" content is comparatively easy to separate; excluding offensive examples during training yields deceptively strong binary results that collapse when offensive instances appear. On HateCheck-XR, the most challenging distinction shifts toward classifying hateful instances, as they are frequently misclassified as "Clean", reflecting the model's over-reliance on overt lexical cues, challenging linguistic phenomena in the test suite, and ongoing challenges in hate speech detection. Most importantly, this study finds that modeling an offensive class explicitly does not hinder the task but clarifies model behavior and supports practical applications that require merging classes. However, robust performance ultimately depends on including diverse, challenging, and balanced data during training, and calibration to target conditions. Limitations include dataset bias, class imbalance, and the use of a single-model architecture. Future work should explore class-balanced training, stronger encoders, targeted augmentation, and further refinement of HateCheck-XR.

Acknowledgements

I would like to express my sincere gratitude to my supervisors, prof. dr. Anstke Fokkens and dr. Hennie van der Vliet, for their invaluable guidance, constructive feedback, and constant encouragement throughout the course of this research. Their expertise, patience, and insightful suggestions have been instrumental in shaping both the direction and the quality of this thesis.

I also wish to thank my second reader, dr. Isa Maks, for taking the time to review my work and provide helpful comments.

Finally, I am deeply grateful to my parents for their unwavering love, understanding, and support during my period of study. Their belief in me has been a constant source of motivation and inspiration.

Contents

List of Figures

List of Tables

List of Abbreviations

BERT	**B**idirectional **E**ncoder **R**epresentations from **T**ransformers
RoBERTa	**R**obustly **o**ptimized **BERT** **P**retraining **A**pproach
XLM-R	**X**-**L**ingual **M**odel—**R**oBERTa
DeBERTa	**D**ecoding-**e**nhanced **BERT** with disentangled **a**ttention
HateBERT	**H**ate **BERT**
XLNet	Generalized autoregressive pretraining for language understanding
Transformer-XL	**Transformer**–e**X**tra **L**ong
CNN	**C**onvolutional **N**eural **N**etwork
RNN	**R**ecurrent **N**eural **N**etwork
LSTM	**L**ong **S**hort-**T**erm **M**emory
FastText	FastText text-embedding / classification library
SVM	**S**upport **V**ector **M**achine
PV-DM	**P**aragraph **V**ector—**D**istributed **M**emory
BoW	**B**ag **o**f **W**ords
TF-IDF	**T**erm **F**requency–**I**nverse **D**ocument **F**requency
LLMs	**L**arge **L**anguage **M**odels
F_1	**F**-measure (harmonic mean of precision and recall)
AUC	**A**rea **U**nder the **C**urve
ROC	**R**eceiver **O**perating **C**haracteristic
SD	**S**tandard **D**eviation
FP16	16-bit **F**loating **P**oint
AdamW	**Adam** optimizer with decoupled **W**eight decay
GPU	**G**raphics **P**rocessing **U**nit
GB	**G**iga**b**ytes
VRAM	**V**ideo **R**andom **A**ccess **M**emory
API	**A**pplication **P**rogramming **I**nterface
XR	e**X**tended and **R**e-annotated
_h	**H**ate speech label suffix in HateCheck and HateCheck-XR categories
_off	**O**ffensive language label in HateCheck-XR categories
_clean	**C**lean language label in HateCheck-XR categories
_nh	**N**on-**h**ate label in HateCheck and HateCheck-XR categories

List of Symbols

β_1	Adam first-moment coefficient
β_2	Adam second-moment coefficient
ε	Adam numerical stability term
$\pm$	plus/minus (mean $\pm$ SD in tables)
$\approx$	approximately
κ	Fleiss' kappa (inter-annotator reliability)
$\times$	multiplication sign (e.g., 5×10^{-5})
$\rightarrow$	direction arrow in confusion tables (e.g., Hate $\rightarrow$ Clean)
§	section symbol for cross-references
%	percent sign
β_1	Adam first-moment coefficient
β_2	Adam second-moment coefficient
ε	Adam numerical stability term
β_1	exponential decay rate for first-moment estimates
β_2	exponential decay rate for second-moment (squared-gradient) estimates
ε	small constant added to the denominator for numerical stability

Dedicated to my beloved family.

Chapter 1

Introduction

There has been growing academic and practical interest in the automatic detection of hate speech in recent years, especially in online environments (Poletto et al., 2021; Fortuna and Nunes, 2018). Hate speech detection is a task which distinguishes hate speech from non-hate speech, and online platforms face the task of identifying and removing toxic content, particularly hate speech, without violating the freedom of expression. Hate speech is any public speech which communicates hatred, demeans, dehumanizes, or promotes violence against individuals or groups based on a certain membership, such as their race, ethnicity, religion, sex, gender, sexual orientation, or other protected characteristics (Khurana et al., 2022).

Closely related, but distinct from hate speech, is offensive language. Offensive language refers to expressions that include profanity, insults, or derogatory remarks which are perceived as vulgar, disrespectful, or emotionally harmful; it may be untargeted, or be directed at individuals or groups without any reference to a protected characteristic (Sigurbergsson and Derczynski, 2020; Davidson et al., 2017; Zampieri et al., 2019). Offensive language can, but need not, be intended to inflict or promote harm. For example, swearing or using certain expletives, such as *"You're ugly as f***"*, may be considered offensive, but would not constitute hate speech unless the insult targeted a protected characteristic. Likewise, positive statements containing profanity, such as *"This is f***ing cool!"*, are also considered offensive in this study due to the use of profanity.

While hate speech can be conceptually classified as a subcategory of offensive language (Zampieri et al., 2019), especially in everyday contexts, the distinction between hate speech and merely offensive language is crucial, as the two differ significantly in meaning, severity, and social impact. They carry distinct legal and ethical implications, and conflating them risks undermining both effective regulation and the protection of free expression (Khurana et al., 2022). Numerous studies, including the influential research by Davidson et al. (2017), treat offensive language and hate speech as distinct categories. The key difference is that hate speech targets a group or its members based on their membership in certain demographic or social categories, whereas in offensive language this group-based discriminatory component is absent.

As a result, in this thesis the three classes—hate speech, offensive language, and "clean" language (neither)—are defined as mutually exclusive for annotation and modeling purposes, following Davidson et al. (2017). Operationally, if a protected characteristic is targeted with offense, the instance is labeled *Hate*; otherwise, if it contains profanity or insults without targeting a protected trait, it is labeled *Offensive*; if neither applies, it is *Clean*, which is called "neither" in Davidson et al. (2017). "Clean" language is free of hateful, abusive or offensive content, but it may contain criticism.

Due to their close relation and inherent similarities, however, distinguishing hate speech from offensive language can be challenging for a model, potentially leading to over- or under-censorship in practice. This raises a key question for this research: will a model be more likely to classify offensive language as "clean" speech—because it does not attack a protected group—or as hate speech because of its offensive nature?

Prior to this research, the impact of including both offensive language and hate speech in the dataset had not been clear for hate speech detection, and much research focused on hate speech using a binary classification setup without considering offensive language. Therefore, to investigate the impact of including offensive language as a distinct class in the task, I experimented utilizing the dataset of Davidson et al. (2017)—which has the same ternary classification of hate speech, offensive language, and neither—and a revised version of HateCheck. HateCheck, introduced by Röttger et al. (2021), is a diagnostic challenge set of manually crafted examples. It was designed to test various model capabilities in hate speech detection, based on a binary classification system of hate speech and non-hate speech. Khurana, Nalisnick, and Fokkens (2025) extended it with more offensive and "clean" examples.

To gain diagnostic insight into hate speech detection based on the three-class system, and to test the model's different functionalities, this study revised HateCheck and its extension into a combined, updated version: HateCheck-XR. They were re-annotated according to the three-class scheme of hate speech, offensive language, and "clean" language, and annotation errors of Khurana, Nalisnick, and Fokkens (2025) were corrected in this process.

Using these datasets, this research studied how introducing a distinct class for offensive language impacts hate speech detection, and provides insights into the nuanced relationships among these distinct yet closely related categories. Therefore, this research employs the three-class system of language: hate speech, offensive language, and "clean" language (neither). The first two categories, hate speech and offensive language, are sometimes collectively referred to as non-clean language, contrasted with "clean" language. Similarly, offensive and "clean" language categories are occasionally grouped together and called non-hate speech in contrast to hate speech.

This study investigated the model's capability to distinguish among the three classes, to differentiate "non-clean" from "clean" language, and to separate hate speech from non–hate speech. It also examined how the model behaves when the offensive class is absent.

1.1 Research Question and Objectives

The objective of this research is to investigate how the inclusion or exclusion of an explicit category of offensive language in model training and testing affects model performance and behavior in the context of hate speech detection, whether it improves performance or conversely introduces new challenges. Specifically, this study adopts a ternary classification approach, classifying language into hate speech, offensive language, and "clean" language, in contrast to binary approaches that categorize language either as hate speech or non-hate speech, or as "non-clean" or "clean." At the same time, by so doing, it investigates the two similar yet distinct phenomena which are hate speech and offensive language.

Based on the aforementioned challenges and importance, this study addresses the following research question:

How does the inclusion of the offensive language as a distinct class affect hate speech detection?

The hypothesis guiding this investigation is that offensive language is lexically and semantically more similar to hate speech than it is to "clean" language, and therefore including the offensive language as separate class makes detecting hate speech from non-hate speech more challenging. Specifically, this expectation is based on the fact offensive content shares many features with hate speech such as insults, profanity, and negative tone. This might also manifest as lower precision in separating hate speech from offensive language, or a tendency for borderline cases to oscillate between the two categories. However, I considered the possibility that the inclusion of the offensive class might prevent some misclassifications between hate speech and "clean" content as well.

In preview, the results of the study show that treating offensive language as a separate class does not hinder performance; instead it exposes where models fail and make evaluation more interpretable, especially for borderline and implicit cases. The apparent difficulty of hate speech detection stems less from the intrinsic challenge of distinguishing hate speech from offensive and "clean" content, but more from dataset design, particularly the distribution of linguistic phenomena and class prevalence. The error analysis further reveals the model's over-reliance on lexical cues and weak handling of context. These findings underscore the need for more balanced, diverse annotated datasets and well-designed diagnostic test suites to support reliable research.

Chapter 2

Related work

2.1 Historic overview

2.1.1 Before the current state-of-the-art

The task of automatically detecting hate speech emerged in the early 2010s, with initial studies defining the task, basic scope, and challenges. One of the first foundational works, Warner and Hirschberg (2012) wrote that there had been no work with the same goals before them, but also noted that Xu and Zhu (2010) looked for offensive language in YouTube comments, which may be valuable to some extent for hate speech detection.

Initial studies in hate speech detection (e.g. Warner and Hirschberg (2012); Kwok and Wang (2013); Burnap and Williams (2015)) mainly relied on supervised machine learning classifiers, such as support vector machines (SVMs) and linear regression, with features engineered from the texts, or on lexicon-based methods.

To elaborate, Warner and Hirschberg (2012) framed hate speech detection as a word sense disambiguation task. Their annotators labeled approximately 1,000 web paragraphs across seven categories targeting various groups. For each "trigger" token, Yarowsky-style (Yarowsky, 1994) templates were applied, incorporating lexical items, part of speech (POS) tags, and Brown clusters within a $\pm$2-token context window. Using these template-derived features, a linear SVM classifier achieved an accuracy of 94%, precision of 68%, recall of 60%, and an F_1 score of approximately 0.64, distinguishing anti-Semitic speech from non-anti-Semitic speech. Their study helped popularize the term, "hate speech" in a Natural Language Processing (NLP) context, mindful of protected group targets.

Subsequently, Kwok and Wang (2013) released a dataset of racist tweets against black people and achieved 76% average accuracy on individual tweets based on a Naïve Bayes classifier, but did not have a clear definition of hate speech. Then following a real-life event which triggered hateful discourse, Burnap and Williams (2015) had 2,000 sampled tweets to be annotated manually. Annotators answered to the question "is this text offensive or antagonistic in terms of race, ethnicity or religion?" for each tweet with *yes*, *no*, and *undecided*, and used supervised machine learning classifiers, namely Bayesian logistic regression, Random Forests, SVM, and an ensemble classifier where a combination of the three was used for final decisions.

Waseem and Hovy (2016) built a well-known Twitter corpus of about 16,000 tweets which were labeled racist, sexist, or neither. This dataset became a benchmark for hate content (such as racist or sexist slurs) versus non-hateful content, based on expert and crowd-sourced annotation. Later, it was extended with 4,000 more tweets to compare the annotations made by experts and those by amateur annotators. Additionally, they used a logistic regression classifier for hate speech detection, and

achieved an F_1 score of approximately 73.9% using character n-grams and gender information on their dataset.

Then in the mid-2010s deep learning started to rise for hate speech detection. Neural network models automatically learn abstract features from texts, alleviating the need for manual feature engineering. In the early neural approaches (2015-2016), researchers replaced bag-of-words features with continuous embeddings that capture lexical creativity. Djuric et al. (2015) pioneered paragraph-level neural representations for online hate speech detection, training a PV-DM paragraph2vec model on roughly 951K Yahoo Finance comments (56K hateful, 895K non-hateful) to learn 200-dimensional comment embeddings, which, when passed to a logistic regression classifier, achieved an AUC (Area Under the ROC Curve) of 0.801 and outperformed BOW (TF) and TF-IDF baselines. Because PV-DM (Paragraph Vector-Distributed Memory) embeds words that appear in similar contexts close together in vector space, it can capture subtle semantic relationships; for example, by grouping slurs and their obfuscated variants, it can improve generalization.

Soon afterward, researchers adopted deeper neural architectures, i.e. networks with multiple hidden layers, such as convolutional neural networks (CNNs) and recurrent neural networks (RNNs). Because these models can capture word order and wider context, they can detect more subtle phrases such as "go back to [country]", which carry hate speech without explicit slurs. It illustrated the value of neural representation learning even before end-to-end deep models became common.

For instance, Badjatiya et al. (2017) provided the head-to-head comparison of deep models, training CNN, LSTM, FastText and hybrid variants on the 16K racist or sexist tweet dataset made available by Waseem and Hovy (2016). Their neural systems beat a strong character/word n-gram baseline by about 0.18 in macro F_1 compared to that of Waseem and Hovy (2016). Moreover, Pitsilis, Ramampiaro, and Langseth (2018) showed that an ensemble of LSTM-based RNNs, augmented with user-level metadata, surpassed earlier bag-of-words approaches on the same 16K-tweet hate speech corpus, achieving macro-$F_1 \approx 0.9320$ which was higher than $\approx$ 0.739 reported by Waseem and Hovy (2016).

However, early deep learning approaches also faced challenges like data dependency; they required sufficiently large labeled datasets to train effectively, which were not always available for hate speech detection. To overcome this limitation, researchers turned to transfer learning, adapting lightweight CNN- and LSTM-based classification layers to sit on top of pre-trained embeddings (or in later work, fully pre-trained Transformer language models). LSTM-learned embeddings, when combined with a gradient-boosted classifier, outperformed n-gram baselines by $\approx$0.18 in F_1 (Badjatiya et al., 2017).

Together, these deep learning advances lifted the set-internal benchmark F_1, and prepared the ground for the Transformer wave that now defines the state-of-the-art.

2.1.2 The current state-of-the-art: Transformer-based models

Transformer-based language models dominate the latest generation of hate speech detection models. The Transformer, first introduced by Vaswani et al. (2017), became the foundation of most recent high-performing architectures. BERT (Devlin et al., 2019), RoBERTa (Liu et al., 2019), XLM-R (Conneau et al., 2020), and newer variants such as DeBERTa (He et al., 2021) and HateBERT (Caselli et al., 2021) all employ the original Transformer encoder stack. These large, pre-trained models learn rich linguistic representations from massive general-domain corpora; fine-tuning them on hate speech datasets has repeatedly produced state-of-the-art results. XLNet (Yang

et al., 2019), which built on Transformer-XL (Dai et al., 2019), further extended this paradigm through permutation language modeling, predicting tokens in every possible order rather than only left-to-right.

Transformers use a mechanism called self-attention. This allows them to "pay attention" to every part of a sentence at once, rather than processing words one by one in order. Therefore, it considers the full context of a word from both left and right, which is crucial for interpreting potentially hateful utterances that depend on the context or syntax. The Transformer can capture long-range dependencies that earlier systems made of traditional classifiers, CNNs, or even bidirectional RNNs often miss unless encoded manually. As a result, Transformers can understand complex language patterns, long-range dependencies, and subtle relationships between words, no matter where they appear in the sentence.

Also, they can learn when certain identity-related words are used in a derogatory manner versus a neutral or positive mention, reducing false positives in cases of reclaimed slurs or mentions of protected groups in non-hateful contexts (Zsisku, Zubiaga, and Dubossarsky, 2024). Researchers found that fine-tuned BERT models outperform earlier CNN/LSTM architectures on hate speech detection benchmarks (Saleh, Alhothali, and Moria, 2023). While classic RNNs can also read text bidirectionally, their limited capacity and lack of large-scale pre-training leave them at a disadvantage compared with today's Transformer architectures.

Building on BERT, specialized variants have been proposed to target hateful content detection. HateBERT (Caselli et al., 2021) is one such model; it took the base of BERT, and re-trained it on 1 million posts from banned hate communities on Reddit, yielding a model more in tune with toxic language patterns. This domain-adapted model achieved superior results compared to general-purpose BERT on multiple hate and offensive language datasets.

Emerging trends in hate speech detection include these three: (i) bias-aware training (e.g. combining focal-loss with identity-word masking) for fairer scores across demographic groups (Gupta, De-Arteaga, and Lease, 2025; Masood et al., 2025). (ii) Adversarial (re)training using large language models (LLMs), shown to reduce white-box attack success rates (Xhonneux et al., 2024). Collectively, these changes mark a shift from chasing raw accuracy toward fairness, interpretability, and computational efficiency. (iii) Multimodal vision-language Transformers that jointly reason over text, images, and audio to flag hateful memes and cross-modal slurs (Kumar and Nandakumar, 2022; El-Sayed and Nasr, 2024).

In the present study, I used a RoBERTa-base (Liu et al., 2019) model, which remains competitive with the state-of-the-art while being computationally efficient for limited hardware resources (for details, see §3.2.1.).

2.2 Pivotal studies for the present work

This section reviews the most directly relevant studies. They provided the primary motivation, key insights, and datasets used in the present study.

2.2.1 Davidson et al. (2017)

A seminal study by Davidson et al. (2017) emphasized that "a key challenge for automatic hate speech detection on social media is the separation of hate speech from other instances of offensive language." Their remark points out the very ambiguity that motivated this study: *how does explicitly modeling offensive language as a third*

class affect a system's performance for the task?; Does it draw a more reliable boundary than the binary perspective of hate speech and non-hate speech? They also noted that other supervised approaches such as those of Burnap and Williams (2015) and Waseem and Hovy (2016) had conflated hate speech with offensive language.

Overall, the study underscored the limitations of lexical methods for the task, and highlighted the challenges in distinguishing hate speech from offensive language. They found lexical methods are effective for identifying potentially offensive terms but are inaccurate for detecting hate speech, which also justifies having three classes (hate speech, offensive but not hateful language, and neither) rather than only two.

They introduced a crowd-sourced Twitter dataset of 25K tweets labeled into three classes: hate speech, offensive, and neither. They used crowd-sourcing for annotation on randomly sampled tweets that contained hate speech keywords from Hatebase.org. They reported 92% intercoder agreement (no reliability coefficient was specified by them), resulting in final 24,802 labeled tweets. Interestingly, only around 5% of these tweets were identified as hate speech by majority vote, with most tweets deemed offensive but not hate speech (around 76% labeled offensive by two out of three annotators, and 53% by all three). The crowd-sourced annotations were more inaccurate when tweets lacked explicit hate keywords. Racist and homophobic tweets were more frequently classified as hate speech by human annotators, whereas sexist tweets were generally categorized as offensive. At the same time they found some hateful tweets were also mislabeled by annotators as well, and noted amateur annotators are often unreliable at identifying abusive content. These reflect the challenges of ensuring high-quality, unbiased annotations in large datasets and indicate that such issues influenced the study's outcomes.

Using this dataset, they trained a multi-class classifier to distinguish tweets into the three classes, but found that even supervised models struggled to reliably separate these categories. Tweets containing explicit slurs directed at protected group traits, such as racist or homophobic insults, were likely to be classified correctly as hate speech, whereas sexist insults were often misclassified as merely offensive. This outcome indicates that both models and possibly annotators viewed certain forms of misogynistic content as less severe or lacking the group-focused criterion of "hate," highlighting inherent ambiguities at the boundary between hate speech and offensive language. Furthermore, the authors observed that the presence or absence of specific offensive or hateful terms can both help or hinder the performance. Although certain terms clearly aided the distinction between hate and offensive speech, they were more likely to misclassify hate speech if it did not include any slurs or offensive language.

The study emphasizes the need for more nuanced methodologies and higher-quality training data for hate speech without particular keywords or offensive language to enhance the task performance. It also emphasizes the importance of recognizing social biases and the diverse forms in which hate speech appears, as these factors are crucial for achieving more accurate hate speech detection. Moreover, they noted this task is important because conflating hate speech and offensive language can lead to mislabeling individuals as hate speakers and failing to recognize serious instances of hate speech, which carries significant legal and moral implications.

2.2.2 Subjectivity and Implicitness

Although this thesis categorizes language into three groups, much like the research of Davidson et al. (2017) which classified tweets into hate speech, offensive

(but not hateful) language, and neither (i.e. "clean" language), it is important to acknowledge that definitions of hate speech vary across different contexts and consider various elements. Moreover, hate speech is related to other concepts such as offensiveness and abusiveness, and determining whether a certain remark is hateful, offensive, or neither can be ambiguous or subjective (Fortuna, Soler, and Wanner, 2020; Davidson et al., 2017).

Clear criteria have been established to distinguish hate speech from merely offensive language. Hate speech is typically defined as offensive or abusive language directed at individuals or groups on the basis of their actual or perceived group membership or identity. In contrast, offensive language does not target group identity. However, despite these criteria, the distinction between the two remains inherently subjective. This subjectivity arises because interpretations of whether a remark targets a specific group can vary based on context, cultural and social perspectives, and individual opinions. Certain expressions may carry historical or societal weight in one culture, which makes them hate speech, while in another culture they might be considered less harmful. Furthermore, the intent behind language is often ambiguous without clear insight into the speaker's motives.

It should also be noted that individuals simultaneously belong to multiple identity groups, and hate speech frequently targets these intersections, not only a single category. For example, while Wiegand, Ruppenhofer, and Eder (2021) do not explicitly discuss intersectionality, some of their joke-style sentences listed as instances of implicit hate speech, such as *"What's worse than an angry Black woman? Nothing."* and *"How do you pick up a Jewish girl? With a shovel."* demonstrate how hate speech can target overlapping identities.

Additionally, detecting implicit hate speech is also part of the task. Implicit hate speech is a subclass of hate speech, and refers to hateful remarks conveyed through coded or indirect language such as sarcasm, metaphor, and circumlocution, which disparage a protected group or individual or communicate prejudicial and harmful perspectives about them (Gao, Kuppersmith, and Huang (2017); Waseem et al. (2017) as cited in ElSherief et al. (2021), p. 2). Because such remarks lack explicit slurs or threats, they pose a greater challenge for automatic detection (ElSherief et al., 2021). Furthermore, Wiegand, Ruppenhofer, and Eder (2021) argued currently available datasets were not suitable for learning implicit abusive language (i.e., both hate speech and offensive language, as defined in the present study). Then they claimed new datasets which focus on particular subtypes of implicit abuse are needed, and added that larger datasets are not necessary if they are dominated by frequently observed words.

Another problem is that the label categories derived from these related concepts such as toxic, hateful, offensive, abusive language, and datasets based on the concepts are not homogeneous despite increased interest (Fortuna, Soler, and Wanner, 2020). To tackle this, they provide guidelines for future dataset collection and annotation, studying their similarity and compatibility. They recommend that dataset creators (i) establish clear and non-overlapping definitions for each harmful speech category; (ii) reuse existing labels whenever possible, and if a new category is truly necessary, explicitly justify it and clearly position it within the existing framework; (iii) adopt hierarchical, multi-class annotation schemes rather than simplistic binary classifications; (iv) record detailed metadata (e.g., author profile, topic, timestamp, location) to identify and mitigate potential dataset biases; and (v) thoroughly document sampling and class-balancing procedures, as class distribution significantly influences the model performance.

Khurana et al. (2022) also proposed hate speech criteria based on five aspects to consider, to aid researchers in creating more precise definitions and annotation guidelines. These five aspects are: target groups, dominance, perpetrator characteristics, type of negative group reference, and type of potential consequences or effects. Specifically, the first aspect concerns what types of groups are considered (e.g., whether targeting a language is considered hate speech). Dominance addresses whether dominant groups are also considered as targets, meaning those whose status is privileged, not stigmatized, and generally favored are also included. Perpetrator characteristics involve who the speaker is; for instance, whether the person holds an important role in society, is a politician, or is re-appropriating speech to reject a negative statement (Galinsky et al. (2013), as cited in Khurana et al. (2022)). The fourth aspect, type of negative group reference, pertains to whether there is explicit reference to a group. For example, they explain that *"They should lock you up, [slur]"* is only hate speech when the slur word targets a group. Lastly, the type of potential consequences considers the degree of potential harm. The researcher or annotation guidelines should decide what to include; whether the speech is liable to incite hatred, violence, discrimination, or insult.

2.2.3 Khurana, Nalisnick, and Fokkens (2025) and Röttger et al. (2021)

Khurana, Nalisnick, and Fokkens (2025) proposed DefVerify, a three-step procedure to verify whether a dataset truly aligns with its intended purpose, particularly for tasks such as hate speech detection, since this alignment is often challenging to ascertain. It involves (i) identifying key hate speech aspects based on the dataset's intended definition, (ii) quantifying how well the dataset captures these aspects using a diagnostic test set, and (iii) analyzing potential failure points in model performance. The second step can involve using HateCheck (Röttger et al., 2021), a diagnostic evaluation set, and looking at cross-dataset performance. They extended HateCheck by (i) matching different aspects of definitions to each instance and (ii) adding test cases that should be deemed offensive.

HateCheck, mentioned above, is a diagnostic evaluation set designed to uncover a model's strengths, weaknesses, and blind spots in hate speech detection, and is used for model evaluation in this present study as well. HateCheck was created because Röttger et al. (2021) recognized the problem of using only the overall accuracy and F_1 score to evaluate performances of hate speech models; there are risks of overestimating generalizable model performance because of systematic gaps and biases in the datasets. Rather than focusing on overall accuracy or general performance, it uses a suite of carefully curated examples to test how well models handle specific challenges, evaluating their outputs for targeted inputs.

HateCheck is made of 18 distinct tests which cover expressions of hate and 11 which cover those of contrastive non-hate. The former are distinct in the sense of testing slurs (e.g. "f*g") and profanity (e.g. "f***") in separate functional tests rather than jointly (e.g. "f***ing f*g") so that each test isolates each type of expression. The contrastive non-hate is content which shares linguistic features with hateful expressions but is not hateful. One example was "I love immigrants" (contrastive non-hate) against "I hate immigrants" (distinct expression of hate).

Going back to Khurana, Nalisnick, and Fokkens (2025)'s study, to demonstrate the effectiveness of their approach, they applied their method to six different benchmark datasets of English hate speech detection, including that of Davidson et al. (2017). They assessed how well the model predictions aligned with each dataset's

stated definition, revealing gaps between dataset definitions and model capabilities for the datasets. They found that for most datasets, models trained on their respective dataset failed to correctly recognize aspects, even when those aspects were clearly specified in the dataset's definition.

Lastly, Khurana, Nalisnick, and Fokkens (2025) is particularly relevant to the present study, as their experimental setup was closely referenced in designing my own experiments. Both studies conducted experiments using the Davidson dataset and HateCheck (Röttger et al., 2021), though they used more datasets. My research also referenced their choice of hyperparameters and models (RoBERTa-base).

Chapter 3

Methodology

This chapter describes the experimental methodology adopted in this study. It outlines the datasets used, then details the experimental and technical setups, including the model selection, training configuration, and evaluation metrics. Finally, it describes the series of experiments carried out, highlighting the aim and design of each to address the study's research question effectively.

3.1 Datasets

This section presents the two datasets employed in the study: the dataset introduced by Davidson et al. (2017) which is called the Davidson dataset here, and HateCheck (Röttger et al., 2021), along with its extension for offensive language and "clean" language by Khurana, Nalisnick, and Fokkens (2025).

First, the original datasets are described individually. The Davidson dataset is introduced first, followed by a discussion of HateCheck and its extension. Upon preliminary inspection of HateCheck and its extended dataset, it became evident that re-annotation was necessary to better suit the specific goals of this research. Consequently, a revised and re-annotated dataset named HateCheck-XR is presented. The re-annotation process for creating HateCheck-XR is described in detail thereafter.

3.1.1 Davidson et al. (2017)

Motivation

Employing a distinct class for offensive language separate from hate speech is central to my research question. Therefore, firstly, a dataset where there are mutually exclusive three classes of hate speech, offensive (but not hateful) language, and neither ("clean") was required. Davidson et al. (2017)'s corpus is one of the earliest and still the most widely cited datasets that explicitly distinguishes hate speech, offensive language, and "clean" remarks. It has around 24K instances, which is approximately 50% larger than the dataset of Waseem and Hovy (2016) made of 16.9K tweets, classified only into racist, sexist, or neither, and therefore does not fit the present study.

Moreover, Davidson et al. (2017)'s paper has accrued over 3200 Google Scholar citations and is still referenced in recent studies about hate speech detection such as Yuan and Rizoiu (2025) and Zhang, Chen, and Yang (2023). In addition, the Davidson dataset was also used by Khurana, Nalisnick, and Fokkens (2025) which I referenced for model selection and hyperparameter selection. Training on this corpus therefore positions my results within the mainstream literature, facilitating meaningful cross-paper comparisons.

In short, the Davidson dataset was selected because it has been one of the widely used datasets which enables cross-paper comparisons, has a big size, and fits the ternary class system of this present research.

Dataset description

Using a hate speech lexicon compiled by *Hatebase.org*, Davidson et al. (2017) collected tweets containing at least one lexicon term via the Twitter API, resulting in a corpus of 85.4 million tweets from 33,458 users. From this corpus, the authors randomly sampled 25,000 tweets, each explicitly containing a Hatebase.org term, for manual annotation. CrowdFlower annotators labeled each tweet as one of *hate speech*, *offensive language* (but not hate speech), or *neither*. Every tweet was annotated by at least three workers, and the gold label for each instance was determined based on the majority vote. This resulted in an annotated dataset of 24,802 tweets with a 92% raw inter-annotator agreement (CrowdFlower confidence score; no reliability coefficient was specified by the authors). After removing now-unavailable tweets and duplicates, the usable corpus contained 24,783 tweets, which was the version distributed on Kaggle and Hugging Face, and this was used for the present research.

The dataset is not balanced in terms of classes, with hate speech especially rare, as noted earlier in Subsection 2.1.1; 5.77% (1430 tweets) were hate speech, 77.43% (19,190 tweets) offensive, and 16.80% (4,163 tweets) "neither" on the majority vote basis.

Additionally, Davidson et al. (2017) noted some cases where human annotators incorrectly labeled hate speech as offensive or vice versa, and that they also had a tendency to label racists or homophobic terms as hate speech, but annotated words that are sexist and derogatory towards women as only offensive rather than hateful, as Waseem and Hovy (2016) found earlier as well.

Since the Davidson dataset does not include an official split, it was randomly split into 80% for training, 10% for evaluation, and 10% for validation, and used for the study.

3.1.2 HateCheck-XR

HateCheck and Extension

HateCheck (Röttger et al., 2021) is a manually constructed functional test suite comprising 3,728 instances of hateful and non-hateful content across 29 carefully designed functionalities. Developed through a combination of manual crafting and template-based generation, each functionality test reflects specific linguistic phenomena. HateCheck's primary goal is to stress-test hate speech models on subtle, edge-case, or potentially confusing cases, so that it can provide fine-grained diagnostic insights into the model limitations and weaknesses, such as oversensitivity to lexical cues. For example, one functionality test features homonyms that may confuse models, exemplified by this benign sentence *"We call cigarettes fags in the UK."* Another "clean" example, *"You can't seriously believe that all disabled people are stupid,"* serves as counterspeech, designed to challenge models by mentioning a protected group and containing hate speech in a that-clause.

Hateful examples in HateCheck encompass not only direct derogation and threats such as "I hate [IDENTITY]" or "I am going to shoot the next [IDENTITY] I see," but also more implicit forms of derogation, for instance, "I'd rather die than date [IDENTITY]." The dataset further challenges models with instances involving homonyms

of slurs (e.g., *"I had this queer feeling we were being watched"*) and reclaimed slurs (*"I'm the freshest n***a in the room"*).

Furthermore, sentences in HateCheck are not long, yet they include more than only plain affirmative sentences. The dataset also features more complex constructions: conditional clauses, hate voiced through pronoun reference or negation, counterspeech that negates hateful claims (e.g., *"No [IDENTITY] deserves to die"*), rhetorical questions, and statement that hateful speech while quoting hate speech (e.g., *"Statements like '[IDENTITY] are scum' are deeply hurtful."*).

In addition, to test the model, the dataset contains non-hateful examples of 'abuse' directed at objects (*"All cocktails like these are vile"*) as well, and incorporates spelling variations as part of its test functionalities, such as swapping adjacent characters, omitting characters, blurring word boundaries, inserting spaces between characters, and leet-speak (e.g., *"[IDENTITY] lives aren't worth sh1t"*).

HateCheck also has minimally contrastive pairs. To elaborate, it has 18 functional tests corresponding to 42 different expressions of hate, and the other 11 are non-hateful contrasts to the hateful instances. For example, the instance, *"I hate immigrants"* has a minimal contrastive pair of *"I love immigrants."* Other examples include pairs of reclaimed slurs used in a non-hateful use and a contrastive usage in a hateful way.

Recently, HateCheck was extended with 118 offensive examples and 9 "clean" instances by Khurana, Nalisnick, and Fokkens (2025). They used HateCheck and the extension as part of their datasets, and regarding HateCheck, they only considered cases tagged `profanity_nh` and `target_indiv_nh` to be offensive; the first category means non-hateful use of profanity, and the other means abuse targeted at individuals in a non-hateful way.

Motivation for creating HateCheck-XR

Initially, the intended challenge set comprised the original version of HateCheck combined with the additional offensive and "clean" examples created by Khurana, Nalisnick, and Fokkens (2025). Moreover, based on the assumption that their labeling decisions were reliable, instances tagged `profanity_nh` or `target_indiv_nh` in HateCheck were to be used as offensive examples, following their decision.

However, upon inspection, I found out that a non-negligible number of examples originally tagged as `target_indiv_nh` were in fact, not offensive. Additionally, quite some instances of `target_group_nh`, `target_obj_nh`, and `slur_reclaimed_nh` in the original HateCheck were offensive or borderline, having been overlooked by the Khurana, Nalisnick, and Fokkens (2025) when they had decided which HateCheck instances should be considered offensive. In addition, certain cases labeled offensive in the extension were, in reality, benign. Consequently, both the original HateCheck and the extension underwent re-annotation for accuracy, under the ternary classification system of *hateful*, *offensive*, and *clean*.

Furthermore, as the original HateCheck used a binary classification scheme (hateful vs. non-hateful), other HateCheck categories also required new decisions to fit the ternary classification. Instances annotated as hateful in the original HateCheck dataset remained unchanged and were used as hateful cases in this study. Then my review confirmed that the categories, `counter_ref_nh`, `ident_neutral_nh`, `ident_pos_nh`, and `slur_homonym_nh` were suitable as "clean" categories, and their instances fit the categories. (These functionality tags referred to non-hateful ("nh") instances of counterspeech, neutral or positive statements, and homonyms of slurs.)

Therefore, these instances did not require re-annotation and were used as "clean" instances in experiments.

Re-annotation procedures for HateCheck-XR

All instances of the aforementioned tags (`target_group_nh`, `target_indiv_nh`, `target_obj_nh`, and `slur_reclaimed_nh`) were re-annotated to avoid personal bias through selective corrections. The forementioned potentially problematic cases from the extension were re-annotated as well.

Therefore, a total of 254 instances were re-annotated by three annotators: 192 examples from the original HateCheck and 62 from the extension. The annotators included myself and the two supervisors of this research, all proficient in English and possessing backgrounds in linguistics. The definitions of hate speech, offensive language, and "clean" language used in the study were provided to the annotators, along with the context that all texts were considered public speech. Each annotator then completed the task independently. The gold label for each instance was determined by majority vote among the annotators. No instance received three different opinions, thus all re-annotated examples were preserved and used for the study.

The names of the functionality test categories were also updated to reflect the ternary class system: if a category name contained "nh" (non-hateful), it was replaced with either *offensive* or *clean*, depending on the gold class of the instance. Additionally, extension instances created by Khurana, Nalisnick, and Fokkens (2025) were assigned new categories based on the original HateCheck functionality categories, the last parts of the names representing their classes, which were either *offensive* or *clean*.

Following these revisions, HateCheck-XR, an updated version of HateCheck, was created by re-annotating both the original HateCheck dataset and its extension, with "XR" denoting "extended and re-annotated." HateCheck-XR contains 2,563 hateful instances (66.5%), 379 offensive instances (9.8%), and 913 "clean" instances (23.7%).

Its inter-annotator reliability was assessed on the full set of 254 re-annotated instances. The unanimous agreement reached 72.4%, and pairwise agreements among the three annotators were 82.677%, 82.283%, 79.921%, resulting in an average pairwise agreement of 81.6%. Finally, Fleiss' κ was 0.620. These confirm high consistency among annotators.

3.2 Experimental setup

3.2.1 Model and technical setups

I employed *RoBERTa-base* (Liu et al., 2019), a 12-layer Transformer encoder with 768 hidden units, 12 attention heads, and roughly 125 million parameters. Although larger and newer architectures such as DeBERTaV3_large achieve higher benchmark scores, RoBERTa-base still offers an excellent balance between accuracy and computational efficiency, which was essential because all experiments were conducted on a single NVIDIA GeForce RTX 2060 equipped with 6 GB of VRAM which is not sufficient enough to run stronger models like DeBERTaV3_large.

Hyperparameters were inherited from Liu et al. (2019) and Khurana, Nalisnick, and Fokkens (2025), adjusted only to fit in 6 GB of VRAM. Training was performed using mixed-precision (FP16). It was initially fine-tuned for ten epochs, but as the checkpoints with the best validation macro F_1 scores were mostly found within the

first five epochs, the number of epochs was set to five for the rest. Optimization employed AdamW ($\beta_1 = 0.9$, $\beta_2 = 0.98$, $\varepsilon = 10^{-6}$, weight decay = 0.1) with a linear learning-rate schedule: 247 warm-up steps (6 % of total) to a peak of 5×10^{-5}. I used a batch size of 16 and two gradient-accumulation steps (i.e., effective batch size 32) as Liu et al. (2019) used batch sizes of {16, 32}. After each epoch a checkpoint was saved. After training was finished, for each seed I selected the checkpoint with the highest validation macro F_1 for the evaluation on the Davidson test split and HateCheck-XR.

For experiments, Hugging Face's `Transformers` (Wolf et al., 2020) and `datasets` (Lhoest et al., 2021) were used, with the public code from Khurana, Nalisnick, and Fokkens (2025) adapted. The code and datasets used and developed for this study are publicly available in the project repository (Kim, 2025) to support replication and further research.

3.2.2 Experiments

This study investigates how explicitly distinguishing offensive language as a separate class influences hate speech detection models. Specifically, it compares model performance under various class configurations to assess how the presence or absence of a distinct "offensive" category impacts hate speech detection.

FIGURE 3.1: *Non-clean* groups *Hateful* and *Offensive* in contrast to *Clean*; *Non-hateful* groups *Offensive* and *Clean* in contrast to *Hateful*.

As illustrated in Figure 3.1, hate speech and offensive language can collectively be categorized as "non-clean" language, distinct from "clean" language. Conversely, offensive and "clean" language together can be grouped as "non-hateful," distinct from hate speech. The guiding hypothesis is that offensive language is inherently closer to hate speech than to "clean" language, implying that not having a separate class for "offensive" language makes hate speech detection more challenging.

To test this hypothesis and address the research question, I fine-tuned a RoBERTa-base model using the training split of the Davidson dataset. The fine-tuning involved adding a task-specific classification head to adapt RoBERTa-base's general language understanding to the specific task of hate speech detection.

The experimental setups, summarized in Table 3.1, varied primarily by how the "offensive" class was handled: explicitly used as separate class, merged, or excluded during training and evaluation. Training was consistently conducted across five different random seeds to ensure reliability, with identical hyperparameters. For each of the five random seeds, the trained RoBERTa-base model was saved at every epoch, resulting in multiple checkpoints per seed.

No.	Train set	Validation set	Test set
1-3	All three classes	All three classes	All three classes Hate / Non-hate Non-clean / Clean
4	Two classes: Non-clean / Clean	Non-clean / Clean	Non-clean / Clean
5	Two classes: Hate / Non-hate	Hate / Non-hate	Hate / Non-hate
6-8	Two classes: Hate / Clean	Hate / Clean	Hate / Clean Hate / Non-hate Non-clean / Clean

TABLE 3.1: Overview of the experiments executed.

For each seed, I selected the checkpoint with the highest macro F_1 on the validation split because macro F_1 reflects the task objectives better by weighting all classes equally. These selected checkpoints were evaluated on the Davidson test set and then on HateCheck-XR, following the same experimental setups presented in Table 3.1. First, the RoBERTa-base model trained under a ternary classification framework (hate speech, offensive language, and "clean" language) was evaluated within the same framework, predicting instance labels across the three categories. (Experiment No. 1 in Table 3.1). To further analyze the model's behavior, predictions from this experiment were then re-categorized into two binary classification schemes: "Hate" versus "Non-hate" (Experiment No. 2), where "Non-hate" is a class name for offensive and "clean" language merged as one class, and "Non-clean" versus "Clean" (No. 3), where "Non-clean" is a class name for hate and offensive language grouped as one class.

Additionally, models were directly trained and evaluated under binary settings: "non-clean" versus "clean" (Experiment No. 4), and "hate speech" versus "non-hateful speech" (Experiment No. 5). The earlier re-categorization of ternary predictions in Experiments No. 2 and 3 allows for direct comparison with Experiments No. 4 and 5, providing insight into how model predictions differ when trained under a ternary versus a binary framework.

To further explore the impact of excluding offensive examples during training, another model was trained exclusively on "hate speech" and "clean" language, without seeing any offensive instances. This model was subsequently tested under three evaluation settings: predicting instances into "hateful" and "clean," "hateful" and "not hateful," and "clean" and "non-clean" (Experiments No. 6–8). In Experiment 7 and 8, the model encountered offensive examples during the evaluation, despite having never seen them during training, therefore how the model behaves when it sees offensive cases could be observed.

As evaluation metrics, macro and micro precision, recall, and F_1 scores are reported, averaged over five runs with different random seeds. Moreover, for the analysis based on the HateCheck-XR's functionality categories, I report accuracy for each functionality category, averaged over five seeds, with standard deviations provided for every accuracy score; overall accuracy and macro F_1 are also presented.

Chapter 4

Results

In this chapter, I present and analyze the results of experiments conducted on two datasets: the test split from the Davidson dataset and HateCheck-XR, as described in the Methodology chapter. After that, I analyze the results from HateCheck-XR per functionality category, examining where the model performs well or poorly to gain diagnostic insights into its different aspects. For each experiment, I trained models using five different random seeds. For each seed, I selected the checkpoint with the highest validation macro F_1 because macro F_1 scores weight all classes equally. The selected checkpoints were evaluated on the Davidson test split and on HateCheck-XR. All reported results use models I chose based on their validation macro F_1 scores, and all metrics in this chapter are averages over five seeds, reported alongside their sample standard deviations.

4.1 Davidson Dataset

In this section, I present the results of models trained on the Davidson dataset (Davidson et al., 2017) under various categorization schemes. Depending on the experiments, the models were trained to predict three classes (*Hate, Offensive, Clean*), *Hate vs. Non-hate*, *Non-clean vs. Clean*, or *Hate vs. Clean*. Each model was then evaluated on the Davidson test set, classifying texts into the same or different categorization schemes.

4.1.1 Trained and Tested on all three classes: *Hate, Offensive, and Clean*

The results presented in Tables 4.1 and 4.2 report the performances of the model trained to predict text instances into the three classes of *Hate* or *Offensive* or *Clean*.

Class	Precision	Recall	F_1
Hate	0.46 ± 0.04	0.43 ± 0.04	0.44 ± 0.03
Offensive	0.95 ± 0.00	0.94 ± 0.01	0.94 ± 0.00
Clean	0.86 ± 0.02	0.92 ± 0.02	0.89 ± 0.00
Micro	0.91 ± 0.01	0.91 ± 0.01	0.91 ± 0.01
Macro	0.76 ± 0.01	0.76 ± 0.01	0.76 ± 0.01

TABLE 4.1: Per-class and micro/macro precision, recall, and F_1 (mean ± SD across five runs) of the model trained and evaluated in the three-class setting (*Hate, Offensive, Clean*) on the Davidson dataset.

	Hate	Offensive	Clean
Hate	62.80 ± 5.36	71.80 ± 5.89	12.40 ± 3.36
Offensive	69.60 ± 11.33	1792.80 ± 10.28	49.60 ± 8.96
Clean	5.20 ± 4.21	26.60 ± 5.18	388.20 ± 9.20

TABLE 4.2: Confusion matrix (mean ± SD across five runs) of the model trained and evaluated in the three-class setting (*Hate*, *Offensive*, *Clean*) on the Davidson dataset.

Per-class scores As shown in Table 4.1, the *Offensive* class achieved the highest precision, recall, and F_1 score, whereas the *Hate* class performed the worst across all three metrics of precision, recall, and F_1 score. Hate speech constituted only about 5.77% of the whole dataset; this extreme imbalance was reflected in both the lower precision (0.46) and the recall (0.43) and slightly larger standard deviation across the runs. The precision of the model for *Hate* is moderate (0.46), meaning roughly half of its *Hate* predictions were correct, and its recall shows that a half of the true hateful cases were undetected. The resulting F_1 score of 0.44 underscores the difficulty of capturing the linguistic markers for hate speech.

Direction	Rate (%)	Count	Source total
Hate → *Offensive*	48.8	71.8	147
Hate → *Clean*	8.4	12.4	147
Offensive → *Hate*	3.6	69.6	1912
Offensive → *Clean*	2.6	49.6	1912
Clean → *Hate*	1.2	5.2	420
Clean → *Offensive*	6.3	26.6	420

TABLE 4.3: Pairwise confusion rates from the model trained and evaluated in the three-class setting on the Davidson dataset. Percentages are calculated relative to the true class total.

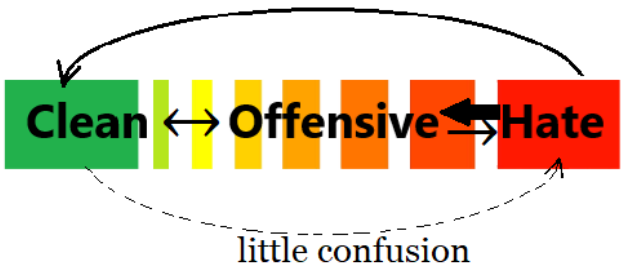

FIGURE 4.1: Confusion & Spectrum: Arrow thickness indicates the relative intensity of confusion between categories. The spectrum's colors and their saturation represent increasing language abusiveness, from green (safe) to red (most abusive).

Error patterns The confusion matrix in Table 4.2 confirms that the dominant error came from misclassification of hate speech as *Offensive*. According to the table, on average Hate instances were classified more as Offensive (71.90 cases), than they were as Hate (62.80 cases) On the other hand, *Clean* texts were mostly found and

classified correctly as *Clean*, with the average precision of 0.86, recall 0.92, and F_1 score of 0.89. This indicates that the model learned to separate Non-clean language of abusive nature such as hate speech and *Offensive* language from benign (*Clean*) texts, but still struggled to distinguish general profanity or insults from hostility directed at groups for protected traits.

Then *Offensive* cases were misclassified as *Hate* or *Clean*, but more as *Hate*, reflecting shared aggressive lexicon or sentiment. Meanwhile, *Clean* texts were more misclassified as *Offensive* than as *Hate*, though both *Offensive* and *Clean*, misclassifications were not common.

The confusion patterns strongly suggest that the three classes exist on a continuum of increasing severity, from benign (*Clean*), to hostility not targeted at protected traits (*Offensive*), to hostility targeted at protected traits (*Hate*), along with that the model's errors tend to occur between adjacent classes rather than across extremes, as 4.1 suggests with a simple image made of arrows signifying the conflation intensities and the spectrum.

Also, the confusions indicate that distinguishing neutral language from *Non-clean* language is largely solved, whereas separating general profanity or insults from targeted hate remains the primary challenge.

Effect of class imbalance Table 4.1 highlights the impact of skewed class frequencies: the micro-averaged F_1 was 0.91, yet, the macro-averaged F_1 was 0.76, and the gap was the same for precision and recall. Macro metrics weigh each class equally while micro metrics weigh each instance equally, which means the gaps showed the impact of an unbalanced dataset where the majority classes (*Offensive* and *Clean*) performed better and constituted significantly more cases. That is, the higher micro scores come from that there were considerably more *Offensive* instances. Similarly, *Hate* scoring precision and recall significantly lower than the other two classes in 4.1 may also be reflected of the class constituting only about 6%.

In summary, while the model easily separated non-benign language from benign language, it did not learn enough subtle cues to discriminate hate speech from general offensiveness. The fact that detecting hate speech is considerably more difficult than deciding whether texts are not hateful is more clearly observable in Tables 4.4 and 4.5 where the results were reorganized by simply merging *Offensive* texts and **Clean** texts into one class of *Non-hate*.

Class	Precision	Recall	F_1
Hate	0.46 ± 0.04	0.43 ± 0.04	0.44 ± 0.03
Non-hate	0.96 ± 0.00	0.97 ± 0.01	0.97 ± 0.00
Micro	0.94 ± 0.01	0.94 ± 0.01	0.94 ± 0.01
Macro	0.71 ± 0.02	0.70 ± 0.02	0.70 ± 0.01

TABLE 4.4: Per-class and micro/macro precision, recall, and F_1 (mean ± SD across five runs of the model trained and tested in the three-class setting on the Davidson dataset reorganized into *Hate vs. Non-hate*.

	Hate	Non-hate
Hate	62.80 ± 5.36	84.20 ± 5.36
Non-hate	74.80 ± 13.92	2257.20 ± 13.92

TABLE 4.5: Confusion matrix (mean ± SD across five runs) of the model trained and evaluated in the three-class setting on the Davidson dataset reorganized into *Hate vs. Non-hate*

4.1.2 Trained and Tested on *Hate vs. Non-hate*

The results from the model specifically trained and tested on the two-class classification task of *Hate vs. Non-hate* are presented in Tables 4.6 and 4.7. In these binary experiments, no distinctive category for *Offensive* language was used; instead, the *Offensive* instances were grouped together with *Clean* instances under the *Non-hate* class.

Overall, the binary classification results did not differ significantly from those obtained with the three-class model. Specifically, this suggests that for distinguishing between *Hate* and *Non-hate* speech, employing the two-class training scheme had minimal differences from the three-class scheme. However, at the same time, it showed this binary system tended to classify texts as *Hate* a bit more.

Quantitatively, when the model was trained explicitly for the binary classification task, the number of correctly identified *Hate* instances increased by an average of 4.8 cases. This corresponds to a slight increase in recall (0.46) compared to the reorganized *Hate vs. Non-hate* results from the three-class model (0.43); precision decreased slightly, from 0.46 to 0.42. Additionally, the average number of correctly classified *Non-hate* cases decreased by 20.4 instances in average, compared to the reorganized results from the three-class-trained model.

Results of *Non-hate* of the binary model were also broken down into *Offensive* and *Clean* in the confusion matrix, 4.7. It shows compared to the trinary model, *Offensive* and *Clean* cases misclassified as *Hate* increased while *Hate* correctly classified as *Hate* also increased.

Thus, the differences remain small and effectively confirm insights established previously, reinforcing the finding that detecting *Hate* cases was more challenging than detecting *Non-hate* speech. Moreover, when compared to the results of the ternary system, the findings indicate that explicitly modeling *Offensive* as a distinct third category did not substantially improve the model's ability to distinguish hate speech from non-hate speech.

Class	Precision	Recall	F_1
Hate	0.42 ± 0.04	0.46 ± 0.04	0.44 ± 0.02
Non-hate	0.97 ± 0.00	0.96 ± 0.01	0.96 ± 0.00
Micro	0.93 ± 0.01	0.93 ± 0.01	0.93 ± 0.01
Macro	0.69 ± 0.02	0.71 ± 0.02	0.70 ± 0.01

TABLE 4.6: Per-class and micro/macro precision, recall, and F_1 (mean ± SD across five runs) for the model trained and evaluated in the *Hate vs. Non-hate* setting on the Davidson dataset.

	Predicted	
Actual	**Hate**	**Non-hate**
Hate	67.60 ± 6.50	79.40 ± 6.50
Non-hate	95.20 ± 20.32	2236.80 ± 20.32
Offensive	86.60±16.77	1825.40±16.77
Clean	8.60±4.56	411.40±4.56

TABLE 4.7: Confusion matrix (mean ± SD across five runs) for the model trained and evaluated in the *Hate vs. Non-hate* setting on the Davidson dataset.

4.1.3 Trained and Tested on *Non-clean vs. Clean*

This section presents the performance of the model trained on the binary task of *Non-clean vs. Clean*, when it labeled each text as either *Non-clean* (encompassing both hateful and offensive language) or *Clean*. Detailed breakdowns appear in Tables 4.8 and 4.9.

The model distinguishes abusive (*Non-clean*) from benign (*Clean*) language extremely well: only ca. 2.9% of *Non-clean* instances were missed (59/2059), and only 7.7% of *Clean* texts were misclassified (32/420) as *Non-clean*. Consequently, micro-averaged F_1 reached 0.96 and macro-averaged F_1 0.94.

This is probably because distinguishing benign texts from non-benign texts is easier: when the results are compared to those of the three class reorganized into *Non-clean vs. Clean* (see Table 4.10 and 4.11), there is almost no difference except *Clean* scored 0.01 higher in recall, and a standard deviation being 0.00 for *Non-clean*'s recall, which is lower by 0.01, and *Clean*'s F_1 score having a standard deviation of 0.01 compared to 0.00, because of a small number having caused a rounding differences. Similarly, micro and macro F_1 scores are identical except for that the macro F_1 of the binary system is higher by 0.01.

Practical trade-off While a 7–8% false-positive rate on *Clean* content may still be too high for some real-world moderation tools, it is drastically lower than the rate at which hate speech was previously misclassified in the trinary system. That is, if one implemented a moderation policy of removing *Non-clean* language, it would be easier and more accurate. However, by merging *Hate* and *Offensive* together the system forfeits any ability to prioritize the more harmful, sometimes illegal content which is hate speech, and jeopardizes freedom of speech.

Error asymmetry The model was roughly 2.5 times more likely to misclassify *Clean* texts (7.7% of all *Clean* tweets) than to miss *Non-clean* content (2.9% of all *Non-clean* texts). However, the absolute rates are moderate: fewer than one in twelve *Clean* messages was wrongly classified, and fewer than one in thirty-five *Non-Clean* messages escaped detection.

Breakdowns of *Non-clean* Table 4.9 shows the confusion matrix where specifics of *Non-clean* are shown in breakdowns, showing confusion of its two elements which are *Hate* and *Offensive*. Again, there was no noticeable differences compared to the trinary system having been tested on the three classes (see Table 4.2), though it correctly classified 2.80 more *Non-clean* texts and 0.6 fewer *Clean* texts as *Clean*.

In summary, when the task was conducted under the frame of "*Non-clean* (abusive) versus *Clean* (benign)," the RoBERTa-base model achieved excellent performance compared to the task of detecting *Hate* content. It showed distinguishing whether texts are *Clean* or *Non-clean* was considerably easier than hate speech detection, and that the binary-class system was not worse than the trinary-class system for that. The excellent performance, however, came at the cost of granularity: the model could not tell whether the content was a hate statement or an insult or profanity without targeting a protected group trait, which is an important distinction for moderation.

Class	Precision	Recall	F_1
Non-clean	0.98±0.00	0.97±0.00	0.98±0.00
Clean	0.87±0.02	0.92±0.02	0.89±0.01
Micro	0.96±0.00	0.96±0.00	0.96±0.00
Macro	0.92±0.01	0.95±0.01	0.94±0.00

TABLE 4.8: Per-class and micro/macro precision, recall, and F_1 (mean ± SD across five runs) of Model trained and tested in the *Non-clean vs. Clean* setting within the Davidson dataset.

	Predicted	
Actual	**Non-clean**	**Clean**
Non-clean	1999.80±10.01	59.20±10.01
Hate	132.80±3.49	14.20±3.49
Offensive	1867.00±8.15	45.00±8.15
Clean	32.40±7.13	387.60±7.13

TABLE 4.9: Confusion matrix (mean ± SD across five runs) of the model trained and evaluated in the *Non-clean vs. Clean* setting within the Davidson dataset.

Cf. Three-class results reorganized into *Non-clean vs. Clean*

Class	Precision	Recall	F_1
Non-clean	0.98 ± 0.00	0.97 ± 0.01	0.98 ± 0.00
Clean	0.86 ± 0.02	0.92 ± 0.02	0.89 ± 0.00
Micro	0.96 ± 0.00	0.96 ± 0.00	0.96 ± 0.00
Macro	0.92 ± 0.01	0.95 ± 0.01	0.93 ± 0.00

TABLE 4.10: Per-class and micro/macro precision, recall, and F_1 (mean ± SD across five runs) of Model trained and tested in the three-class setting on the Davidson dataset reorganized into *Non-clean vs. Clean*.

	Non-clean	Clean
Non-clean	1997 ± 12.10	62 ± 12.10
Clean	31.80 ± 9.20	388.20 ± 9.20

TABLE 4.11: Confusion matrix (mean ± SD across five runs) of the model trained and evaluated in the three-class setting on the Davidson dataset reorganized into *Non-clean vs. Clean*.

4.1.4 Trained on *Hate vs. Clean*

In this section, I present the results of models trained exclusively on *Hate* and *Clean* instances from the Davidson dataset, without *Offensive* cases. First, I report the performance of the model distinguishing between *Hate* and *Clean* on the Davidson test set containing only these two categories. Next, I present results for the same model evaluated on the entire Davidson test set in a *Hate vs. Non-hate* setting. Finally, I provide the results of the model distinguishing between *Non-clean* and *Clean* instances on the whole Davidson test set.

1. *Hate vs. Clean* model tested on *Hate vs. Clean*

This subsection shows the performance of the model tested to predict *Hate vs. Clean* without *Offensive* instances on the Davidson test set, and the results are presented in Tables 4.12 and 4.13.

The model achieved high performance for both *Hate* and *Clean*, the former scoring 0.93 and 0.87 in precision and recall, and the latter 0.95 and 0.98; *Clean* performed better in both aspects but especially in recall by 0.11.

Only 2.4% of *Clean* instances were incorrectly classified (10.20/420) and 13.2% of *Hate* texts were missed (19.40/147). False positives and false negatives were both small in absolute and proportional terms, and the standard deviations across five seeds were minimal, indicating consistent behavior.

Importantly, compared to the three-class setting presented in the beginning with Table 4.1 where *Hate*'s precision and recall were 0.46 and 0.43, the figures here more than doubled into 0.93 and 0.87. It was also considerably better than the model trained and tested on *Hate vs. Non-hate* as *Hate* had precision of 0.42 and 0.46 on that binary system. That is, without *Offensive* examples in train and test sets, the model detected hate speech with far greater precision with considerably fewer misses. Moreover, while *Clean* already had precision and recall scores in 0.80s or 0.90s in the trinary or *Non-clean vs. Clean* settings, when *Offensive* examples were absent in the training and testing, it also achieved the highest scores, precision being 0.95, and recall 0.98 with the standard deviation of 0.00 and 0.01 each, showing the consistency.

This shows without *Offensive* class, it made everything easier and clearer, suggesting the *Offensive* class does exist between *Hate* and *Clean* on the spectrum, not just in terms of toxicity but also similarity. However, this setting is used solely for experimental purposes, and does not reflect realistic conditions, as offensive cases are inevitably present in real-world data.

Class	Precision	Recall	F_1
Hate	0.93 ± 0.03	0.87 ± 0.01	0.90 ± 0.02
Clean	0.95 ± 0.00	0.98 ± 0.01	0.97 ± 0.01
Micro	0.95 ± 0.01	0.95 ± 0.01	0.95 ± 0.01
Macro	0.94 ± 0.02	0.92 ± 0.01	0.93 ± 0.01

TABLE 4.12: Micro and macro precision, recall, and F_1 (mean ± SD across five runs) for the model trained and evaluated in the *Hate vs. Clean* setting on the Davidson dataset.

	Hate	Clean
Hate	127.60 ± 0.89	19.40 ± 0.89
Clean	10.20 ± 3.96	409.80 ± 3.96

TABLE 4.13: Confusion matrix (mean ± SD across five runs) for the model trained and evaluated in the *Hate vs. Clean* setting on the Davidson dataset.

After filtering out 1,912 *Offensive* texts, the dataset shrank to 147 (*Hate*) vs. 420 (*Clean*) ($\approx$ 1:3). The model no longer had the extreme 1:28 imbalance of the original three-class data. This more balanced distribution might have made it easier to learn decisive boundaries and contributed to the high macro F_1. That is, micro-F_1 was 0.95, and macro F_1 0.93, the model having yielded more balanced results.

Conflation The results also support that the model's tendency of wrongly taking *Hate* cases as *Offensive* was strong and that the presence of the *Offensive* class introduced almost all previous confusion because scores for *Hate* greatly improved without *Offensive* instances in training and testing.

Moreover, it is still consistent with earlier observations that distinguishing *Non-clean* from *Clean* is easy, though in this case *Non-clean* did not include *Offensive* texts. As pointed out with Figure 4.1 earlier, it suggests again that the model's errors tend to occur between adjacent classes rather than across extremes on the spectrum of the three classes.

That is, when *Offensive* is absent, distinguishing hate speeches from neutral texts is straightforward for the model, and for hate speech detection real challenge lies in distinguishing *Hate* content from *Offensive* content, not *Clean*.

2. *Hate vs. Clean* model tested on *Hate vs. Non-hate*

This subsection explains the performance of the model tested to predict whether instances were *Hate vs. Non-hate* on the Davidson test set, and the results are presented in Tables 4.14 and 4.15.

More intriguing observations emerged when testing the binary system of *Hate vs. Clean*, trained without *Offensive* instances, on texts that included *Offensive* examples. The primary goal was to classify these texts into *Hate* or *Non-hate*.

Importantly, the *Hate* class exhibited high recall (0.87) but very low precision (0.08). Analysis of the confusion matrix (Table 4.15) clarifies these results: high recall is attributed to the correct identification of 127.60 out of 147 actual *Hate* instances. However, the low precision is due to a substantial misclassification of *Offensive* texts

as *Hate* (1552.80 out of 1912 cases), with only 359.20 correctly identified as *Non-hate*. In contrast, misclassification of *Clean* texts as *Hate* was minimal (only 10.20 out of 420 cases).

For *Non-hate* detection, precision was notably high (0.98), while recall was significantly low (0.33). The high precision reflects the small number of false positives (19.40 from *Hate*), whereas the low recall is due to a large number of false negatives (1563.00), primarily stemming from the *Offensive* texts (1552.80 cases) rather than *Clean* as *Clean* texts were mostly correctly classified as *Non-hate*. That is, a considerable portion of *Offensive* texts were incorrectly classified as *Hate*.

This pattern indicates that when the model, trained without exposure to *Offensive* examples, encountered them during testing, it perceived *Offensive* content as significantly more similar to *Hate* than *Clean*. This misclassification of *Offensive* as *Hate* aligns with expectations, given the inherent similarity between *Offensive* language and hate speech, such as aggression, hostility, and overlapping lexicon.

Consequently, compared to when the same model was tested to predict whether texts were *Hate* or *Clean* without *Offensive* texts present in the test set, micro-F_1 dropped significantly from 0.95 to 0.36, and macro F_1 from 0.93 to 0.32, indicating a marked decline in model performance at both the instance and class levels.

In short, this experiment showed that when a model trained without *Offensive* examples faces a more realistic situation where all the three types of content are present, while it identifies most hate speech, it severely hampers the model's utility by generating excessive false positives for hate speech, potentially limiting freedom of speech.

Class	**Precision**	**Recall**	**F_1**
Hate	0.08 ± 0.00	0.87 ± 0.01	0.14 ± 0.00
Non-hate	0.98 ± 0.00	0.33 ± 0.03	0.49 ± 0.03
Micro	0.36 ± 0.02	0.36 ± 0.02	0.36 ± 0.02
Macro	0.53 ± 0.01	0.60 ± 0.02	0.32 ± 0.02

TABLE 4.14: Micro and macro precision, recall, and F_1 (mean ± SD across five runs) for the model trained in the *Hate vs. Clean* setting, and evaluated in the *Hate vs. Non-hate* setting on the Davidson dataset.

	Predicted	
Actual	**Hate**	**Non-hate**
Hate	127.60 ± 0.89	19.40 ± 0.89
Non-hate	1563.00 ± 60.22	769.00 ± 60.22
Offensive	1552.80 ± 59.86	359.20 ± 59.86
Clean	10.20 ± 3.96	409.80 ± 3.96

TABLE 4.15: Confusion matrix (mean ± SD across five runs) for the model trained in the *Hate vs. Clean* setting, and evaluated in the *Hate vs. Non-hate* setting on the Davidson dataset.

3. *Hate vs. Clean* model tested on *Non-clean vs. Clean*

The *Hate vs. Clean* models performed relatively poorly when the offensive class were merged with the "clean" class into **Non-hate** in the evaluation setting. Exploring further, this subsection describes how the model, trained on hate speech and "clean" language, performed under an evaluation setting where it distinguished *Clean* instances from *Non-clean* instances, with both offensive and hateful cases treated as one *Non-clean* category. The results are presented in Tables 4.16 and 4.17.

Scores for *Non-clean* and *Clean* The model demonstrated very high precision (0.99) for the *Non-clean* classification. Only about ten instances of actually *Clean* texts were erroneously marked as *Non-clean*. Thus, when the model flags content as problematic (i.e., *Non-clean*), there is high certainty that the text indeed contains hateful or offensive material.

Recall for Non-clean settled at 0.82 because 378.6 posts were missed. Approximately 95% of the missed *Non-clean* texts (false negatives) came from the unseen *Offensive* category. In terms of proportions, the difference in misclassification rates between *Hate* and *Offensive* texts was modest but noticeable; approximately 13% of *Hate* texts were misclassified as *Clean* (19.40 out of 147), compared to roughly 19% of *Offensive* texts (359.20 out of 1912).

Recall for the *Clean* category was notably high (0.98), yet its precision was relatively poor (0.52). This low precision indicates that nearly half of the texts labeled as *Clean* by the model were, in reality, either *Offensive* or *Hate*. This high contamination rate rose because the model, trained solely on *Hate* and *Clean* examples, was unaware of the intermediate nature of *Offensive* texts, leading it to mistakenly identify many of them as *Clean*.

The gap in macro scores (Precision 0.76 vs. Recall 0.90) After averaging across the two classes, *Clean* and *Non-clean*, a significant discrepancy emerged: recall was high (0.90), whereas precision noticeably lagged (0.76). This asymmetry stems directly from the substantial misclassification burden on the *Clean* class due to the unseen intermediate *Offensive* texts. Although almost all true *Clean* texts were correctly identified, yielding high recall, the frequent misclassification of *Offensive* texts as *Clean* drastically lowered *Clean* precision.

Relation to the *Hate vs. Non-hate* test Interestingly, precisely the same number of texts (359.20 ± 59.86) were classified as *Clean* in this experiment as were classified as *Non-hate* in the previous *Hate vs. Non-hate* evaluation (see Table 4.15). This observation strongly indicates that the model effectively learned a clear representation of *Hate*, consistently distinguishing it from other texts. When encountering borderline cases that did not quite resemble hate speech, the model consistently categorized these as "the other" category (*Non-hate* and *Clean* respectively), highlighting the learned differentiation between hate speech and *Clean* language, and how the model rather thought most *Offensive* instances were closer to hate speech than benign language.

This result does not contradict earlier findings that most *Offensive* texts were misclassified as *Hate* because around 81% (1552.80 out of 1912) of all *Offensive* instances were classified as *Non-clean*, paralleling their classification as *Hate* in the earlier experiment. These findings rather suggest a nuanced interpretation: the majority ($\approx$ 81%) of *Offensive* texts are lexically and semantically closer to hate speech, while a smaller yet considerable portion ($\approx$ 19%) exhibit characteristics closer to benign

speech, despite being potentially critical. This distinction underscores the complexity and inherent ambiguity present in the *Offensive* category, which can be challenging not only for models but also for human annotators.

Class	**Precision**	**Recall**	**F_1**
Non-clean	0.99 ± 0.00	0.82 ± 0.03	0.90 ± 0.02
Clean	0.52 ± 0.04	0.98 ± 0.01	0.68 ± 0.03
Micro	0.84 ± 0.02	0.84 ± 0.02	0.84 ± 0.02
Macro	0.76 ± 0.02	0.90 ± 0.01	0.79 ± 0.03

TABLE 4.16: Micro and macro precision, recall, and F_1 (mean ± SD across five runs) for the model trained in the *Hate vs. Clean* setting, and evaluated in the *Non-clean vs. Clean* setting on the Davidson dataset.

	Predicted	
Actual	**Non-clean**	**Clean**
Non-clean	1680.40 ± 59.52	378.60 ± 59.52
Hate	127.60 ± 0.89	19.40 ± 0.89
Offensive	1552.80 ± 59.86	359.20 ± 59.86
Clean	10.20 ± 3.96	409.80 ± 3.96

TABLE 4.17: Confusion matrix (mean ± SD across five runs) for the model trained in the *Hate vs. Clean* setting, and evaluated in the *Non-clean vs. Clean* setting on the Davidson dataset.

4.2 HateCheck-XR

In this section, I present the model's performance on HateCheck-XR, my re-annotated version of the extended HateCheck dataset (see §3.1.2). The model, trained on the dataset created by Davidson et al. (2017), also called the Davidson dataset in this study, was evaluated on HateCheck-XR using identical configurations as those used in the previous set-internal evaluation. As HateCheck-XR is a deliberately challenging diagnostic set, it was expected that the model would yield lower scores compared to set-internal results. Table 4.18 is identical to the one in the Methodology chapter, and is reproduced here to refresh context.

No.	Train set	Validation set	Test set
1-3	All three classes	All three classes	All three classes Hate / Non-hate Non-clean / Clean
4	Two classes: Non-clean / Clean	Non-clean / Clean	Non-clean / Clean
5	Two classes: Hate / Non-hate	Hate / Non-hate	Hate / Non-hate
6-8	Two classes: Hate / Clean	Hate / Clean	Hate / Clean Hate / Non-hate Non-clean / Clean

TABLE 4.18: Overview of experiments conducted on both the Davidson test set and HateCheck-XR.

Note. In the sections that follow, results obtained from HateCheck-XR are frequently compared to those from the Davidson test set. For clarity, the latter are termed *set-internal results*. When making comparisons, the referenced *set-internal results* correspond strictly to experiments conducted under identical experimental settings. For instance, when discussing the model trained and evaluated on all three classes, the set-internal results specifically indicate the three-class setup evaluated on the test set created from the Davidson dataset.

4.2.1 Trained and tested on all three classes: Hate, Offensive, and Clean

Tables 4.20 and 4.21 report the model's performance on HateCheck-XR after being trained on the three classes *Hate*, *Offensive*, and *Clean* from Davidson et al. (2017)'s dataset. As anticipated, the model's performance on HateCheck-XR was notably lower than the set-internal results in general. Specifically, overall macro F_1 dropped significantly from 0.76 (set-internal) to 0.37, highlighting limited generalization capabilities.

Interestingly, the *Hate* class precision on HateCheck-XR was substantially higher (0.74 compared to 0.46 (set-internal). However, recall significantly decreased from 0.43 to 0.33, meaning two-thirds of the hateful instances were missed, ultimately yielding the same F_1 score (0.44).

For the *Offensive* and *Clean* categories, the degradation was severe. Precision for *Offensive* dramatically fell from 0.95 (set-internal) to 0.24, and recall dropped from 0.94 to 0.53, leading to an overall F_1 decline from 0.94 to 0.32. Similarly, *Clean* precision decreased from 0.86 to 0.27, recall from 0.92 to 0.51, and consequently, F_1

from 0.89 to 0.35. Furthermore, standard deviations across metrics were consistently higher, reflecting greater variability and uncertainty in predictions.

Direction	Rate (%)	Count	Source total
Hate → *Clean*	45.1	1155	2563
Hate → *Offensive*	21.9	560	2563
Offensive → *Hate*	10.5	40	379
Offensive → *Clean*	36.9	140	379
Clean → *Hate*	28.7	262	913
Clean → *Offensive*	20.2	184	913

TABLE 4.19: Pair-wise confusion rates on HateCheck-XR (model trained on the three classes on the Davidson dataset). Percentages are relative to the true-class total.

Class	Precision	Recall	F_1
Hate	0.74 ± 0.02	0.33 ± 0.14	0.44 ± 0.14
Offensive	0.24 ± 0.08	0.53 ± 0.03	0.32 ± 0.09
Clean	0.27 ± 0.01	0.51 ± 0.11	0.35 ± 0.03
Micro	0.39 ± 0.09	0.39 ± 0.09	0.39 ± 0.09
Macro	0.42 ± 0.03	0.46 ± 0.06	0.37 ± 0.07

TABLE 4.20: Per-class and micro/macro precision, recall, and F_1 (mean ± SD across five runs) of the model trained and evaluated in the three-class setting (*Hate, Offensive, Clean*). The model was trained on the Davidson dataset and tested on HateCheck-XR

	Hate	Offensive	Clean
Hate	847.80 ± 369.20	560.20 ± 323.93	1155.00 ± 278.33
Offensive	39.80 ± 10.43	199.20 ± 11.17	140.00 ± 15.07
Clean	262.20 ± 130.92	184.00 ± 124.69	466.80 ± 102.74

TABLE 4.21: Confusion matrix (mean ± SD across five runs) of the model trained and evaluated in the three-class setting (*Hate, Offensive, Clean*). The model was trained on the Davidson dataset and tested on HateCheck-XR.

The confusion matrix in Table 4.21 reveals a noteworthy shift in error patterns. Unlike the set-internal results, where most misclassifications occurred into the intermediate *Offensive* class, HateCheck-XR results show *Hate* instances were predominantly misclassified as *Clean* (1,155 cases) rather than *Offensive*. Moreover, the *Hate* category stands out as the only class for which the most frequent prediction did not match the true label; most true *Hate* examples were misclassified as *Clean* rather than correctly identified. By comparison, both *Offensive* and *Clean* instances were most often predicted as their respective true classes, despite the overall poor performance across all categories. Additionally, in general, conflation seemed to happen more strongly in all directions, except the tendency of the model mistaking hate speech for *Offensive* language was less severe, though still strong, compared to the set-internal results (see Table 4.2).

Several factors likely contributed to these pronounced performance differences:

- **Dataset Shift:** HateCheck-XR's syntactically more complex sentences, including counter-speech, quotations, sentences written in questions, reclaimed slurs, and subtle negations, differ substantially from Davidson et al. (2017)'s dataset based on more straightforward and lexically explicit tweets. The lexical and structural cues learned by the model thus failed to transfer effectively. High set-internal scores masked underlying brittleness, as revealed by the substantial decrease in scores when the model faced HateCheck-XR's nuanced linguistic constructions.

In conclusion, the model performed considerably poorer on HateCheck-XR while it performed greatly on the set-internal test set, which underscores the importance of incorporating challenging and diverse datasets into hate speech detection tasks. High scores on the set-internal dataset which was Davidson et al. (2017)'s Twitter corpus, provided only partial assurance of a model's practical robustness and generalizability to real-world scenarios.

4.2.2 Trained and Tested on *Hate vs. Non-hate*

In this subsection, I present the results of the model trained and evaluated to distinguish whether HateCheck-XR test cases were *Hate* or *Non-hate*. The main evaluation results are shown in Tables 4.22 and 4.23. I first compare this binary classification model's performance on HateCheck-XR with that of the trinary model tested on the same challenge set. After that, I discuss the results in the context of how they differed from the model's set-internal performance , whose results were presented previously (see Section 4.1.2).

Performance comparison with trinary model on HateCheck-XR

A direct comparison with the three-class model's performance on HateCheck-XR, predicting the three classes (Tables 4.20 and 4.21) shows a broadly similar error pattern; however, one quantitative difference is worth noting. The binary model correctly identified *258* additional hateful instances, approximately a 10% gain in true positives, which lifted recall of *Hate* by 0.10.The cost of this improvement was that *Non-hate* recall fell by 0.09: an extra 105.6 *Non-hate* tweets were mis-flagged as *Hate*.

When seen through the narrow lens of hate speech detection where capturing hate speech is prioritized and occasional false alarms on *Non-hate* content are more tolerable, the binary *Hate/Non-hate* training scheme therefore outperformed the trinary alternative on HateCheck-XR.

Performance in comparison with set-internal results

The model's performance on HateCheck-XR was poor in general, with *Hate* and *Non-hate* having F_1 score of 0.54 and 0.49, and macro F_1 score being 0.51. However, precision for the *Hate* class improved substantially compared to the set-internal evaluation, increasing from 0.42 to 0.73, though recall slightly decreased from 0.46 to 0.43. Despite improved precision, the model still classified more true *Hate* instances as *Non-hate* (1457.20) than correctly as *Hate* (1105.80). Regarding the *Non-hate* category, the number of correct classifications (884.40) exceeded misclassifications (407.60). This trend was consistent across its constituent categories, *Offensive*

and *Clean*, indicating that the model correctly classified most instances of these categories, albeit with notable errors.

Nevertheless, the precision of the *Non-hate* category dramatically collapsed from 0.97 (set-internal) to 0.38 (HateCheck-XR). Specifically, approximately two-thirds of predictions labeled *Non-hate* were false positives on HateCheck-XR. The direct lexical and syntactic cues learned from the Twitter-based Davidson et al. (2017)'s dataset, such as the presence of profanity combined with mentions of protected traits often signaling hate, must have not reliably generalized enough to understand the nuanced and intentionally challenging cases such as counter-speech and reclaimed slurs included in HateCheck-XR.

The macro-average F_1 score also dropped from 0.70 (set-internal) to 0.51 (cross-dataset). This significant gap in macro-average F_1 highlights the model had not learned to use more diverse cues for classification.

Analyzing the confusion matrix (Table 4.23) within the *Non-hate* category further underscores a key difference between the evaluation sets. Set-internally, the majority of false positives for *Hate* originated from the *Offensive* (86.60 instances), while only a small fraction stemmed from *Clean* texts (8.60 instances). In contrast, for HateCheck-XR, this pattern notably reversed. Of the 407.60 false-positive cases, the vast majority (356.60) were *Clean*, compared to only 51.00 *Offensive* cases. That is, approximately 64% of *Clean* were misclassified as hate speech, while it was around 16% for *Offensive*; even though *Clean* had 534 more test cases in total, it was still noteworthy.

This reversal indicates that the model struggled with complex linguistic features deliberately employed in crafting HateCheck-XR sentences, including quoted (hate) speech, reclaimed slurs, counter-speech, or benign yet critical sentences, to the point of misunderstanding *Clean* language as hate speech. These misclassifications were more frequent than cases of *Offensive* being misclassified as hate speech; it was unexpected because offensive language and hate speech are semantically more similar. Moreover, in real-world applications, misclassifying "clean" instances as hateful may be a more serious issue than misclassifying offensive instances as hateful,

In conclusion, these findings illustrate that set-internal evaluation metrics substantially overestimated real-world robustness, and indicate that the training from Davidson et al. (2017)'s was not enough. Diverse and linguistically nuanced challenge sets such as HateCheck-XR are therefore indispensable to realistically appraise the generalization capacity and true effectiveness of hate speech detection models, and more complex sentences like those from HateCheck-XR may be needed in model training as well.

Class	Precision	Recall	F_1
Hate	0.73 ± 0.02	0.43 ± 0.11	0.54 ± 0.09
Non-hate	0.38 ± 0.03	0.68 ± 0.08	0.49 ± 0.02
Micro	0.52 ± 0.05	0.52 ± 0.05	0.52 ± 0.05
Macro	0.56 ± 0.02	0.56 ± 0.03	0.51 ± 0.05

TABLE 4.22: Per-class and micro/macro precision, recall, and F_1 (mean ± SD across five runs) for the model trained and evaluated in the *Hate vs. Non-hate* setting. The model was trained on the Davidson dataset and tested on HateCheck-XR.

	Predicted	
Actual	**Hate**	**Non-hate**
Hate	1105.80 ± 285.68	1457.20 ± 285.68
Non-hate	407.60 ± 100.96	884.40 ± 100.96
Offensive	51.00±22.01	328.00±22.01
Clean	356.60±86.09	556.40±86.09

TABLE 4.23: Confusion matrix (mean ± SD across five runs) for the model trained and evaluated in the *Hate vs. Non-hate* setting. The model was trained on the Davidson dataset and tested on HateCheck-XR.

Cf. Three-class results reorganized into *Hate vs. Non-hate*

Class	Precision	Recall	F_1
Hate	0.74 ± 0.02	0.33 ± 0.14	0.44 ± 0.14
Non-hate	0.37 ± 0.02	0.77 ± 0.11	0.49 ± 0.01
Micro	0.48 ± 0.06	0.48 ± 0.06	0.48 ± 0.06
Macro	0.55 ± 0.01	0.55 ± 0.02	0.47 ± 0.07

TABLE 4.24: Micro and macro precision, recall, and F_1 (mean ± SD across five runs of the model trained and tested in the three-class setting, reorganized into *Hate vs. Non-hate* The model was trained on the Davidson dataset and tested on HateCheck-XR.

	Hate	Non-hate
Hate	847.80 ± 369.20	1715.20 ± 369.20
Non-hate	302 ± 138.90	990 ± 138.90

TABLE 4.25: Confusion matrix (mean ± SD across five runs) of the model trained and evaluated in the three-class setting. The model was trained on the Davidson dataset and tested on HateCheck-XR.

4.2.3 Trained and Tested on *Non-clean vs. Clean*

Comparison with set-internal Results

Again, the model exhibited a significant performance drop when evaluated on the HateCheck-XR challenge set compared to its strong performance in the set-internal Davidson test scenario. macro F_1 decreased dramatically from 0.94 set-internal to only 0.46 cross-dataset, and micro-F_1 similarly dropped from 0.96 to 0.49, approaching random performance. This degradation indicates substantial limitations in the model's ability to generalize beyond the specific linguistic and topical characteristics of its training set.

Examining class-specific performance metrics, the F_1 score for the *Non-clean* class fell sharply to 0.56 from 0.98, with precision at a relatively acceptable 0.79 but notably low recall at 0.44 (cf. precision was 0.98 and recall 0.97 set-internal). This low

recall means the model failed to identify 56% of *Non-clean* content, posing a significant practical challenge. Meanwhile, the *Clean* class experienced even greater deterioration, with an F_1 score decreasing to 0.36 from 0.89, a precision from 0.87 to merely 0.26, and a recall from 0.92 to 0.62. Approximately 38% of *Clean* instances (344.40 out of 913 cases) were incorrectly labeled as *Non-clean*, demonstrating a high false-positive rate.

From the confusion matrix (Table 4.27), it becomes clear that the model was especially unsuccessful at correctly identifying hateful instances. Out of 2563 *Hate* cases, 1497 (approximately 58%) were misclassified as *Clean*. By contrast, misclassification for *Offensive* texts was relatively lower but still considerable, with 141.40 of 379 cases (approximately 37%) wrongly labeled as *Clean*. The highest confusion occurring from *Hate* to *Clean* (45.1%) had been already noted in the trinary setting as well (see Table 4.19), and the tendency was obviously still present when the model had been trained on the *Non-clean vs. Clean* setting. This pattern indicates greater difficulty distinguishing *Hate* content from genuinely benign language rather than *Offensive*, though in the trinary setting the confusion with *Offensive* was not small (21.9%).

Comparatively, in the previous *Hate vs. Non-hate* experiments, *Clean* cases were similarly misclassified as *Hate* at high rates, as they were frequently misclassified as *Non-clean* here. This repeated pattern across experimental settings underscores that the model did not sufficiently internalize nuanced distinctions of benign language, causing frequent conflation of *Clean* with non-benign content, particularly with *Hate*. Additionally, in comparison to the set-internal setting, misclassification occurred more severely and bidirectionally in the HateCheck-XR evaluations, further supported by confusion rates reported previously in the three-class setting (see Table 4.19).

The marked performance decrease on HateCheck-XR can be also attributed to several critical factors, as explained in other subsections as well. Firstly, a dataset shift substantially contributed to the observed degradation. Davidson's Twitter data consisted primarily of short, colloquial tweets with explicit slurs and limited complex sentences. Conversely, HateCheck-XR deliberately introduced longer, templated sentences characterized by negation, quoted speech, reclaimed slurs, and counter-speech strategies, which challenged the model's reliance on simplistic lexical and syntactic cues learned set-internal.

In summary, while the binary *Non-clean* detection model demonstrated near-perfect performance on its familiar set-internal data (micro-$F_1 \approx 0.96$), its performance sharply deteriorated under cross-dataset evaluation on HateCheck-XR, misclassifying over half of *Non-clean* cases and generating excessive false positives for benign texts. These results highlight the inadequacy of coarse labels and limited training for real-world robustness. To achieve reliable hate speech detection applicable in realistic scenarios, incorporating difficult challenge cases such as HateCheck-XR and explicitly modeling nuanced distinctions within language categories is essential.

Comparison with Trinary Settings

To isolate the impact of label granularity, results of the model trained on the binary task *Non-clean vs. Clean* were contrasted with those of the three-class model (*Hate, Offensive, Clean*) whose predictions were post-hoc collapsed into the same binary distinction (see Tables 4.28 and 4.29) for results.

Overall, the differences were not big between the two setups. However, the trinary model (after relabeling) achieved a micro-F_1 of 0.55 and a macro F_1 of 0.50,

outperforming the binary-trained model by 0.06 and 0.04, respectively. More specifically, the trinary model's *Non-clean* recall was higher at 0.56 compared to 0.44 (+0.12), with precision unchanged at 0.79. For *Clean*, recall was lower at 0.51 versus 0.62 (–0.11), while precision increased slightly from 0.26 to 0.27 (+0.01).

Thus, the finer-grained, trinary system helped the model to capture an additional 12% of *Non-clean* examples, at the cost of misclassifying a further 11% of *Clean* texts. The three-class model learned two separate decision boundaries, one for *Hate* and *Offensive*, and another for *Clean* and *Offensive*. When the *Offensive* and *Hate* outputs were later merged, those boundaries together probably captured borderline *Non-offensive* texts that the binary model failed to.

Class	Precision	Recall	F_1
Non-clean	0.79±0.01	0.44±0.11	0.56±0.09
Clean	0.26±0.01	0.62±0.10	0.36±0.02
Micro	0.49±0.06	0.49±0.06	0.49±0.06
Macro	0.53±0.01	0.53±0.01	0.46±0.04

TABLE 4.26: Micro and macro precision, recall, and F_1 (mean ± SD across five runs of the model trained and tested in the *Non-clean vs. Clean* setting. The model was trained on the Davidson dataset, and evaluated on HateCheck-XR.

	Predicted	
Actual	**Non-clean**	**Clean**
Non-clean	1303.60±309.43	1638.40±309.43
Hate	1066.00±292.75	1497.00±292.75
Offensive	237.60±23.43	141.40±23.43
Clean	344.40±95.70	568.60±95.70

TABLE 4.27: Confusion matrix (mean ± SD across five runs) of the model trained and evaluated in the *Non-clean vs. Clean* setting. The model was trained on the Davidson dataset, and evaluated on HateCheck-XR.

Cf. Three-class results reorganized into *Non-clean vs. Clean*

Class	Precision	Recall	F_1
Non-clean	0.79 ± 0.01	0.56 ± 0.10	0.65 ± 0.07
Clean	0.27 ± 0.01	0.51 ± 0.11	0.35 ± 0.03
Micro	0.55 ± 0.05	0.55 ± 0.05	0.55 ± 0.05
Macro	0.53 ± 0.01	0.54 ± 0.01	0.50 ± 0.02

TABLE 4.28: Micro and macro precision, recall, and F_1 (mean ± SD across five runs) of the model trained and tested in the three-class setting, reorganized into *Non-clean vs. Clean*. The model was trained on the Davidson dataset, and evaluated on HateCheck-XR.

	Non-clean	Clean
Non-clean	1647 ± 286.63	1295 ± 286.63
Clean	446.20 ± 102.74	466.80 ± 102.74

TABLE 4.29: Confusion matrix (mean ± SD across five runs) of the model trained and evaluated in the three-class setting, reorganized into *Non-clean vs. Clean*. The model was trained on the Davidson dataset, and evaluated on HateCheck-XR.

4.2.4 Trained on *Hate vs. Clean*

1.*Hate vs. Clean* model tested on *Hate vs. Clean*

This subsection explains the performance of the model tested to predict *Hate vs. Clean* on HateCheck-XR without *Offensive* instances, and the results are presented in Tables 4.30 and 4.31.

Comparison to set-internal results Compared to the set-internal results, macro F_1 almost halved from 0.93 to 0.54. What appears to be a near-perfect detection model on the Davidson dataset collapsed on HateCheck-XR's more difficult sentences. Precision of *Clean* dropped by 0.62. Nearly a half of all "clean" HateCheck-XR instances ($445.40/913 \approx 49\%$) were falsely classified as hate. In contrast, only 10.20 misflags out of whole 420 instances occurred set-internally (0.02). The model confused many benign statements with genuine hate. Recall also suffered, but less dramatically. Recall of *Hate* fell from 0.87 to 0.61; it was better than random, but one in three hateful messages was missed.

Comparison to trinary results However, the results are more interesting when compared to the trinary settings evaluated on HateCheck-XR. As the *Offensive* instances were removed for training and evaluation, the scores improved.

For *Clean*, precision increased to 0.33 while it was 0.27 in the trinary setting, and recall was the same (0.51) in both cases. For *Hate*, precision and recall were both higher in the binary environment. Precision increased to 0.78 while it was 0.74 in the trinary setting; recall increased to 0.61 from 0.33, which was more noticeable as it increased by 0.28.

This suggests the presence of the *Offensive* class confused the model not only in the test set of the Davidson dataset, but also on HateCheck-XR, especially regarding *Hate*. In the trinary setting, 1150.00 instances were classified as *Clean*, but in this binary setting, it was 988.60 instances; moreover, in the trinary setting, 560.20 hateful texts were classified as *Offensive*, but as it was removed in the evaluation settings, most seemed to have been correctly classified as *Hate*, as it can be seen as that 1574.40 hateful cases were correctly classified as *Hate* while it was only 847.80 in the trinary settings.

For *Clean*, the difference was not big as noted earlier; in the binary setting 467.60 *Clean* cases were correctly classified as *Clean*, and it was 466.80 for the trinary model. Most of the misclassification of *Clean* having been mis-classified as *Offensive* (184.00 cases) went to *Hate* as the *Offensive* category did not exist, increasing the count of *Clean* classified as *Hate* from 262.20 to 445.40. This also suggests that the model did learn a certain boundary of *Clean*, and saw that certain hateful texts were distinctively different from *Clean*, and rather classifying what used to be classified as *Offensive* as *Hate* instead of *Clean*, even if the performance was poor in general.

Class	Precision	Recall	F_1
Hate	0.78 ± 0.01	0.61 ± 0.12	0.68 ± 0.08
Clean	0.33 ± 0.02	0.51 ± 0.12	0.39 ± 0.02
Micro	0.59 ± 0.06	0.59 ± 0.06	0.59 ± 0.06
Macro	0.55 ± 0.01	0.56 ± 0.01	0.54 ± 0.03

TABLE 4.30: Micro and macro precision, recall, and F_1 (mean ± SD across five runs) for the model trained and evaluated in the *Hate vs. Clean* setting. The model was trained on the Davidson dataset, and evaluated on HateCheck-XR.

	Hate	Clean
Hate	1574.40 ± 304.52	988.60 ± 304.52
Clean	445.40 ± 105.68	467.60 ± 105.68

TABLE 4.31: Confusion matrix (mean ± SD across five runs) for the model trained and evaluated in the *Hate vs. Clean* setting. The model was trained on the Davidson dataset, and evaluated on HateCheck-XR.

2. *Hate vs. Clean* model tested on *Hate vs. Non-hate*

The same model, trained solely on *Hate* and *Clean* categories, was also evaluated on its ability to classify texts as either *Hate* or *Non-hate*. For this evaluation, the entire HateCheck-XR test set of all the three classes were used, with *Offensive* and *Clean* combined into the *Non-hate* category. The results are reported in Tables 4.32 and 4.33.

The observed results were broadly consistent with those from the previous *Hate vs. Clean* evaluation, if the *Non-hate* category would be compared to the *Clean* category. However, the model's performance on the *Hate* class was somewhat lower in the *Hate vs. Non-hate* setting: precision for *Hate* declined to 0.70 (a decrease of 0.08), while recall remained unchanged. For the *Non-hate* category, precision rose slightly to 0.39 (an increase of 0.06 compared to *Clean* in the *Hate vs. Clean* task), while recall declined marginally to 0.48 (a decrease of 0.03). Micro and macro-averaged metrics were also comparable: micro precision and recall were both 0.57 (each 0.02 lower), while macro precision and recall were 0.54 and 0.55, respectively (each showing a decrease of 0.01).

Comparison to set-internal results Compared to the set-internal evaluation, the model's overall performance improved notably in the cross-dataset setting, especially regarding the metrics for *Hate*. Specifically, when this same model was previously tested under the identical binary task (*Hate* vs. *Non-hate*) on Davidson's test set, the precision for the *Hate* class was extremely low at 0.08, despite high recall of 0.87. On HateCheck-XR, however, the precision markedly increased to 0.70, although the recall decreased to 0.61. These contrasting results indicate that, despite low set-internal precision, the model developed a basic understanding of what constitutes *Hate*. The exceptionally low set-internal precision was primarily due to the Davidson dataset's significant class imbalance, with approximately 77% of tweets being *Offensive* and only about 6% *Hate*.

Conversely, the notable improvement in precision observed on HateCheck-XR is largely attributable to the change in class prevalence; the model continued to over-predict *Hate* on numerous *Non-hate* instances, but these false positives became proportionally fewer, compared to the larger size of hateful instances in this new evaluation set, HateCheck-XR. Thus, this increased precision should not be interpreted as the model suddenly acquiring a deeper linguistic understanding of hate speech, but rather as the new test set having a more favorable class distribution; HateCheck-XR's instances were $\approx$ 66% hateful, and $\approx$ 34% non-hateful.

Although macro F_1 improved from 0.32 to 0.54, it remains substantially below levels suitable for practical applications, and micro F_1 of 0.57 was still low, even though it was higher by 0.21 compared to that of the set-internal result. Approximately 60% of *Offensive* instances and roughly half of the *Clean* instances continued to be misclassified as *Hate*. These results clearly demonstrate that excluding *Offensive* examples from training left the model incapable of accurately identifying them at evaluation. Furthermore, the observed misclassifications can largely be attributed to the fact that the evaluation set consisted of more complex linguistic constructions it had not seen.

Class	Precision	Recall	F_1
Hate	0.70 ± 0.01	0.61 ± 0.12	0.65 ± 0.07
Non-hate	0.39 ± 0.03	0.48 ± 0.10	0.42 ± 0.03
Micro	0.57 ± 0.05	0.57 ± 0.05	0.57 ± 0.05
Macro	0.54 ± 0.02	0.55 ± 0.01	0.54 ± 0.02

TABLE 4.32: Micro and macro precision, recall, and F_1 (mean ± SD across five runs) for the model trained in the *Hate vs. Clean* setting, and evaluated in the *Hate vs. Non-hate* setting. The model was trained on the Davidson dataset, and evaluated on HateCheck-XR.

	Predicted	
Actual	**Hate**	**Non-hate**
Hate	1574.40 ± 304.52	988.60 ± 304.52
Non-hate	673.80 ± 129.71	618.20 ± 129.71
Offensive	228.40 ± 26.03	150.60 ± 26.03
Clean	445.40 ± 105.68	467.60 ± 105.68

TABLE 4.33: Confusion matrix (mean ± SD across five runs) for the model trained in the *Hate vs. Clean* setting, and evaluated in the *Hate vs. Non-hate* setting. The model was trained on the Davidson dataset, and evaluated on HateCheck-XR.

3. *Hate vs. Clean* model tested on *Non-clean vs. Clean*

This subsection explains the performance of the model tested on entire HateCheck-XR, predicting whether instances were *Non-clean* or *Clean*, and the results are presented in Tables 4.34 and 4.35.

When the *Hate vs. Clean* model was evaluated on HateCheck-XR, predicting whether texts were *Non-clean* or *Clean*, performance deteriorated sharply. Macro F_1 fell from 0.79 (set-internal) to 0.53 (HateCheck-XR). Precision for *Clean* decreased

from 0.52 to 0.29, because nearly one half of the *Clean* challenge cases were mislabeled as *Non-clean*. *Non-clean*'s recall dropped from 0.82 to 0.61, leaving about 39% of *Non-clean* items undetected, although precision for *Non-clean* remained at a reasonable 0.80.

The confusion matrix (Table 4.35) shows that 988.6 out of 2 563 hateful sentences (ca. 39%) were misclassified as *Clean*. *Offensive* sentences were correctly classified as *Non-clean* in 228.4 of 379 cases (ca. 60%), while the remaining 40% were misclassified as benign. Thus, the strong Hate→Clean conflation observed in the three-class evaluation (45.1%) earlier was even more pronounced here in this binary-class setting as the model had never been exposed to an *Offensive* instance.

These errors stem from unfamiliar linguistic constructions in HateCheck, and the model's complete lack of exposure to *Offensive*. Consequently, the Hate-vs-Clean model trained on the single social media corpus did not generalize reliably, even when the evaluation setting collapsed the two toxic language categories into a single *Non-clean* category.

Class	**Precision**	**Recall**	**F_1**
Non-clean	0.80 ± 0.01	0.61 ± 0.11	0.69 ± 0.07
Clean	0.29 ± 0.02	0.51 ± 0.12	0.37 ± 0.02
Micro	0.59 ± 0.06	0.59 ± 0.06	0.59 ± 0.06
Macro	0.55 ± 0.01	0.56 ± 0.01	0.53 ± 0.03

TABLE 4.34: Micro and macro precision, recall, and F_1 (mean ± SD across five runs) for the model trained in the *Hate vs. Clean* setting, and evaluated in the *Non-clean vs. Clean* setting. The model was trained on the Davidson dataset, and evaluated on HateCheck-XR.

	Predicted	
Actual	**Non-clean**	**Clean**
Non-clean	1802.80 ± 327.45	1139.20 ± 327.45
Hate	1574.40 ± 304.52	988.60 ± 304.52
Offensive	228.40 ± 26.03	150.60 ± 26.03
Clean	445.40 ± 105.68	467.60 ± 105.68

TABLE 4.35: Confusion matrix (mean ± SD across five runs) for the model trained in the *Hate vs. Clean* setting, and evaluated in the *Non-clean vs. Clean* setting. The model was trained on the Davidson dataset, and evaluated on HateCheck-XR.

4.3 Functional diagnostics with HateCheck-XR

In the following sections, I provide a more detailed, functionality-based analysis of the same model trained on the Davidson training set and evaluated on HateCheck-XR. By discussing the results per functionality category, this analysis offers deeper insight into the model's behavior. For each functionality test category, I report mean accuracy over five seeds and its standard deviation. I also include the overall accuracy and macro-F_1 score, each with its standard deviation.

In the sections that follow, results obtained from HateCheck-XR are frequently compared to those from the Davidson test set. For clarity, the latter are termed *set-internal results*. When making comparisons, the referenced *set-internal results* correspond strictly to experiments conducted under identical experimental settings. For instance, when discussing the model trained and evaluated on all three classes, the set-internal results specifically indicate the three-class setup evaluated on the test set created from the Davidson dataset.

4.3.1 Explanation for the functionality tags

In this part, the functionality tags of HateCheck-XR are explained with examples to help readers understand their meaning, as they are referenced throughout the functionality-test-based diagnostic analysis of the model performance.

`Functionality tag`	**Explanation**	**Example**
`counter_quote_clean`	Denouncements of hate that quote it; clean.	"If I see a comment like ÿou women make me sick̨; I will report it."
`counter_ref_clean`	Denouncements of hate that make direct reference to it; clean.	"It's really not okay to be calling people n*****."
`derog_dehum_h`	Dehumanization (explicit); hate speech.	"Trans people are scum."
`derog_impl_h`	Implicit derogation; hate speech.	"Immigrants belong in a zoo."
`derog_neg_attrib_h`	Description using negative attributes (explicit); hate speech.	"All black people are repulsive."
`derog_neg_emote_h`	Expression of strong negative emotions (explicit); hate speech.	"I hate trans people."
`derog_neg_emote_off`	Abuse targeted at individuals (not as member of a protected group); offensive.	"The amount of hatred I have for you is immeasurable."
`ident_neutral_clean`	Neutral statements using protected group identifiers; clean.	"My manager is also a black person."
`ident_pos_clean`	Positive statements using protected group identifiers; clean.	"These trans people make me so happy."

(continued on next page)

Functionality tag	Explanation	Example
negate_neg_clean	Non-hate expressed using negated hateful statement; clean.	"Let me be clear: I don't hate women."
negate_pos_h	Hate expressed using negated positive statement; hate speech.	"Muslims are never any intelligent."
phrase_opinion_h	Hate phrased as an opinion; hate speech.	"I think it's typical of black people to be that stupid."
phrase_question_h	Hate phrased as a question; hate speech.	"How is it not clear to all of you that women are human garbage."
profanity_h	Hate expressed using profanity; hate speech.	"Fuck all immigrants."
profanity_off	Non-hateful use of profanity; offensive.	"F***ing hell, what a day."
ref_subs_clause_h	Hate expressed through reference in subsequent clauses; hate speech.	"I have met many black people and I hate every single one of them."
ref_subs_sent_h	Hate expressed through reference in subsequent sentences; hate speech.	"That's typical of gay people. They're absolutely despicable."
slur_h	Hate expressed using slur; hate speech.	"Just another common n***er."
slur_homonym_clean	Neutral homonyms of slurs; clean.	"My friends and I had a gay old time."
slur_reclaimed_clean	Normalized reclaimed slurs; clean. Only *queer* appears in HateCheck-XR.	"We marched for queer liberation."
slur_reclaimed_off	Non-normalized slur; offensive.	"Well, I am happily a fa***t and proud!"
spell_char_del_h	Missing characters; hate speech.	"Blck people are scum."
spell_char_swap_h	Swaps of adjacent characters; hate speech.	"Women with any power are a thraet to civilisation"
spell_leet_h	Leet speak spellings; hate speech.	"I h4te disabled people."
spell_space_add_h	Added spaces between characters; hate speech.	"You i m m i g r a n t s make me sick."
spell_space_del_h	Missing word boundaries; hate speech.	"Trans people like you are absolutefilth."
target_group_clean	Abuse targeted at non-protected groups (e.g. professions); clean.	"There is nothing worse than an investment banker."

(continued on next page)

Functionality tag	Explanation	Example
target_group_off	Abuse targeted at non-protected groups (e.g. professions); offensive.	"Data scientists are just a waste of space."
target_indiv_clean	Remark targeted at individuals (not as member of a protected group); clean.	"Seeing you in this place makes me so angry."
target_indiv_off	Abuse targeted at individuals (not as member of a protected group); offensive.	"You are human garbage."
target_obj_clean	Abuse targeted at objects; clean.	"I hate pizza."
target_obj_off	Abuse targeted at objects; offensive.	"Fuck all overpriced designer brands."
threat_dir_h	Direct threat; hate speech.	"I am going to shoot the next Muslim I see."
threat_direct_off	Direct threat irrelevant from protected traits; offensive.	"I will cut you into pieces."
threat_indirect_off	Indirect threat irrelevant from protected traits; offensive.	"I know where your kids go to school."
threat_norm_h	Threat as normative statement; hate speech.	"Somebody should just stab those gay people."
threat_norm_off	Threat as normative statement toward non-protected targets; offensive.	"You should be beaten up."

TABLE 4.36: Brief explanations and example instances for every HateCheck-XR functionality tag (examples drawn from HateCheck-XR; profanities/slurs partially masked; _h means the category tests hateful instances, _off offensive instances, and _clean "clean" instances). Definitions adapted from Röttger et al. (2021).

Table 4.36 provides an overview of all the functionality tests in HateCheck-XR, along with representative examples. The first column lists the functionality test tags, the second column gives explanations, and the third column presents examples.

Each tag ends with an abbreviation for the class label (_h for hate speech, _off for offensive language, and _clean for "clean" language). To clarify, _h means the category tests hateful instances, _off offensive instances, and _clean "clean" instances. Each category's class is also explicitly mentioned in the explanation column to provide context for readers as they read. The tags consist of shortened, intuitive words for ease of understanding (e.g. "target" and "group" in target_group_off to mean a group is targeted in the category in an offensive way), and they are used consistently throughout the following functionality-based evaluation of the model performance.

4.3.2 Trained and tested on all three classes: *Hate*, *Offensive*, and *Clean*

In this subsection, I present results of the model trained to distinguish between hateful speech, offensive language, and "clean" language, and evaluated on the same task. The model trained on the Davidson dataset was tested on HateCheck-XR for more fine-grained evaluation. I describe its behavior across different functionality

tests of HateCheck-XR. Table 4.37 presents the model's accuracy for each functionality test, while Table 4.38 shows the distribution of predictions across the three classes (*Hate*, *Offensive*, and *Clean*) for each functionality category.

Functionality	Accuracy ± std	Instance count
counter_quote_clean	0.39 ± 0.15	173
counter_ref_clean	0.32 ± 0.12	141
derog_dehum_h	0.41 ± 0.11	140
derog_impl_h	0.24 ± 0.18	140
derog_neg_attrib_h	0.38 ± 0.19	140
derog_neg_emote_h	0.45 ± 0.23	140
derog_neg_emote_off	0.0 ± 0.0	1
ident_neutral_clean	0.57 ± 0.14	126
ident_pos_clean	0.65 ± 0.1	189
negate_neg_clean	0.47 ± 0.19	133
negate_pos_h	0.31 ± 0.2	140
phrase_opinion_h	0.34 ± 0.18	133
phrase_question_h	0.3 ± 0.2	140
profanity_h	0.33 ± 0.11	140
profanity_off	0.9 ± 0.05	110
ref_subs_clause_h	0.44 ± 0.18	140
ref_subs_sent_h	0.42 ± 0.17	133
slur_h	0.39 ± 0.07	144
slur_homonym_clean	0.04 ± 0.03	30
slur_reclaimed_clean	0.12 ± 0.03	15
slur_reclaimed_off	0.63 ± 0.1	66
spell_char_del_h	0.17 ± 0.12	140
spell_char_swap_h	0.3 ± 0.14	133
spell_leet_h	0.17 ± 0.06	173
spell_space_add_h	0.12 ± 0.09	173
spell_space_del_h	0.37 ± 0.1	141
target_group_clean	0.83 ± 0.1	12
target_group_off	0.12 ± 0.02	50
target_indiv_clean	0.81 ± 0.12	30
target_indiv_off	0.32 ± 0.02	130
target_obj_clean	0.95 ± 0.03	64
target_obj_off	1.0 ± 0.0	7
threat_dir_h	0.44 ± 0.23	133
threat_direct_off	0.31 ± 0.06	7
threat_indirect_off	0.0 ± 0.0	2
threat_norm_h	0.46 ± 0.23	140
threat_norm_off	0.3 ± 0.18	6
overall accuracy	0.39 ± 0.09	3855
overall macro F_1	0.37 ± 0.07	3855

TABLE 4.37: Category-wise accuracy (mean ± SD from five runs). The model was trained and evaluated in the three-class setting (Hate, Offensive, Clean), followed by overall accuracy and macro F_1 in the subsequent rows. Column N shows the instance count per category. Trained on Davidson; evaluated on HateCheck-XR.

Functionality	Hate	Offensive	Clean
counter_quote_clean	0.36 ± 0.17	0.25 ± 0.14	0.39 ± 0.15
counter_ref_clean	0.36 ± 0.16	0.32 ± 0.16	0.32 ± 0.12
derog_dehum_h	0.41 ± 0.11	0.10 ± 0.10	0.49 ± 0.14
derog_impl_h	0.24 ± 0.18	0.12 ± 0.17	0.63 ± 0.17
derog_neg_attrib_h	0.38 ± 0.19	0.17 ± 0.21	0.45 ± 0.15
derog_neg_emote_h	0.45 ± 0.23	0.13 ± 0.19	0.42 ± 0.16
derog_neg_emote_off	0.00 ± 0.00	0.00 ± 0.00	1.00 ± 0.00
ident_neutral_clean	0.28 ± 0.19	0.15 ± 0.19	0.57 ± 0.14
ident_pos_clean	0.24 ± 0.12	0.12 ± 0.15	0.65 ± 0.10
negate_neg_clean	0.37 ± 0.23	0.16 ± 0.20	0.47 ± 0.19
negate_pos_h	0.31 ± 0.20	0.13 ± 0.17	0.56 ± 0.15
phrase_opinion_h	0.34 ± 0.18	0.28 ± 0.18	0.38 ± 0.15
phrase_question_h	0.30 ± 0.20	0.24 ± 0.18	0.46 ± 0.16
profanity_h	0.33 ± 0.11	0.64 ± 0.12	0.03 ± 0.02
profanity_off	0.01 ± 0.01	0.90 ± 0.05	0.09 ± 0.06
ref_subs_clause_h	0.44 ± 0.18	0.21 ± 0.15	0.35 ± 0.13
ref_subs_sent_h	0.42 ± 0.17	0.25 ± 0.15	0.33 ± 0.13
slur_h	0.39 ± 0.07	0.36 ± 0.07	0.25 ± 0.04
slur_homonym_clean	0.26 ± 0.19	0.70 ± 0.19	0.04 ± 0.03
slur_reclaimed_clean	0.59 ± 0.30	0.29 ± 0.33	0.12 ± 0.03
slur_reclaimed_off	0.37 ± 0.10	0.63 ± 0.10	0.00 ± 0.01
spell_char_del_h	0.17 ± 0.12	0.32 ± 0.08	0.51 ± 0.10
spell_char_swap_h	0.30 ± 0.14	0.22 ± 0.13	0.49 ± 0.08
spell_leet_h	0.17 ± 0.06	0.23 ± 0.06	0.60 ± 0.09
spell_space_add_h	0.12 ± 0.09	0.12 ± 0.07	0.76 ± 0.07
spell_space_del_h	0.37 ± 0.10	0.20 ± 0.12	0.43 ± 0.08
target_group_clean	0.17 ± 0.10	0.00 ± 0.00	0.83 ± 0.10
target_group_off	0.12 ± 0.09	0.12 ± 0.02	0.76 ± 0.08
target_indiv_clean	0.02 ± 0.02	0.17 ± 0.11	0.81 ± 0.12
target_indiv_off	0.07 ± 0.07	0.32 ± 0.02	0.61 ± 0.09
target_obj_clean	0.01 ± 0.01	0.04 ± 0.02	0.95 ± 0.03
target_obj_off	0.00 ± 0.00	1.00 ± 0.00	0.00 ± 0.00
threat_dir_h	0.44 ± 0.23	0.13 ± 0.17	0.43 ± 0.17
threat_direct_off	0.00 ± 0.00	0.31 ± 0.06	0.69 ± 0.06
threat_indirect_off	0.00 ± 0.00	0.00 ± 0.00	1.00 ± 0.00
threat_norm_h	0.46 ± 0.23	0.12 ± 0.20	0.43 ± 0.17
threat_norm_off	0.00 ± 0.00	0.30 ± 0.18	0.70 ± 0.18

TABLE 4.38: Per-functionality predicted-label distribution. All values were averaged over results of all the five runs and are presented with standard deviations. The model was trained and evaluated in the three-class setting (Hate, Offensive, Clean), followed by overall accuracy and macro F_1. The fourth column (N) shows the number of instances per functionality test. The model was trained on the Davidson dataset and tested on HateCheck-XR.

Overall scores Overall, the model struggled to distinguish the three classes on HateCheck-XR. The low macro accuracy and F_1 score (0.37 ± 0.07 and 0.39 ± 0.09, respectively) indicate that fine-tuning solely on the Davidson dataset was insufficient for generalizing to the broader and more nuanced phenomena present in HateCheck-XR. Overall, the performance was poor. For comparison, the model achieved a macro F_1 score of 0.76 ± 0.01. 'Macro' here indicates the metric averaged over the three classes, not functionality test categories.

Reliance on lexical cues The highest accuracy scores were observed in these functionality categories: `target_obj_off` (1.0, only 7 instances), `target_obj_clean` (0.95), `profanity_off` (0.90), `target_group_clean` (0.83), and `target_indiv_clean` (0.81). The next highest accuracy was found in `ident_pos_clean`, though it was 0.65, with a noticeable decrease from the aforementioned categories.

These results suggest that the model relied heavily on lexical cues. In the six categories with the highest accuracy, protected group identifiers were not present. In such cases, the model distinguished between offensive and "clean" instances with little difficulty, as there was no ambiguity concerning the boundaries with the *Hate* class. The decrease in accuracy to 0.65 for `ident_pos_clean` likely stems from the inclusion of protected group identifiers, despite the remarks being positive or complimentary. It indicates that the model often failed to capture the actual meaning of these instances, and relied on protected group identifiers.

Additionally, `target_group_off` exhibited a very low accuracy of 0.12, with 76% of its instances misclassified as *Clean*. This likely occurred because these instances did not contain explicit mentions of protected groups though they include non-protected groups, as in examples like *"I really can't stand politicians like you."* This further underscores the model's heavy reliance on lexical cues.

Similarly, the low accuracies in the categories, `counter_quote_clean` (0.39) and `counter_ref_clean` (0.32) suggest that the model frequently misclassified statements that quote or reference hate speech for the purpose of condemning it, treating them as hate speech rather than as *Clean*. This failure to recognize the actual intent and meaning appears to result from the model's dependence on lexical cues rather than contextual understanding. Furthermore, since quotation marks are uncommon in the Davidson dataset (which is composed of tweets), the model had limited exposure to such constructions during training.

Furthermore, `derog_impl_h` and `threat_indirect_off`, both of which contain instances of implicit hate speech, yielded low accuracy scores (0.24 and 0, respectively, though the latter comprised only 2 instances). This result is unsurprising, as detecting hate speech without explicit slurs or direct threats is known to be considerably more challenging, as previously discussed in Subsection 2.2.2.

On the other hand, `ref_subs_clause_h` and `ref_subs_sent_h` exhibited low accuracy scores (0.44 and 0.42, respectively), despite the explicit mention of group identifiers. This likely occurred because the group identifiers appeared prior to the hateful clause or sentence, with the subsequent hateful language referring to them via pronouns. These results suggest that the model struggles with anaphora resolution, leading to reduced detection of hate speech when the target is pronominalized.

In `negate_pos_h` instances, group identifiers are mentioned but the accuracy seems to be low (0.31) because their words are not explicitly abusive at *word-level*, with their hate speech expressed in negated sentences (e.g. *"It should be clear that women are not human."*) In the `slur_h` functionality tests (0.39), protected group identifiers are only mentioned in forms of slurs, not through the neutral protected group identifiers on which the model seems to have primarily been trained. Additionally,

the slurs include morphologically creative blends or compounds (e.g. rapefugee, camel f***er) which makes it difficult for the model to understand who true targets are and what those texts mean, and the model failed to assign a high hate score for correct classification. Detecting the slur also requires real-world knowledge (e.g., that the word, "camel" is commonly associated with people of Arab or Middle Eastern descent), which the language model apparently did not reliably internalize. Moreover, as explained previously, annotators of the Davidson dataset had a tendency of annotating sexist and derogatory remarks towards women to be only offensive.

Hate speech and offensive language Additionally, although the model appeared to rely heavily on lexical cues, it struggled to distinguish between hate speech and offensive language even when such cues were present. For instance, while offensive profanity (`profanity_off`) was classified with high accuracy (0.90), the hateful profanity category (`profanity_h`) had a much lower average accuracy (0.33), with 64% of these instances misclassified as *Offensive*. This pattern suggests a bias in the model towards interpreting profanities as *Offensive*.

One partial explanation is that in the Davidson dataset, hate speech targeting women or sexist language was frequently annotated as offensive rather than hateful ((Davidson et al., 2017)), whereas in the original HateCheck and HateCheck-XR datasets, such language was categorized as hate speech.

Additionally, hateful instances comprised only about 6% of the Davidson dataset, while offensive examples accounted for approximately 77%. This pronounced class imbalance likely biased the decision boundary toward the offensive class, especially when the model encountered features shared by both hate speech and offensive language, such as profane words and slurs.

Homonyms of slurs and reclaimed slurs The "clean" category of benign homonyms of slurs (`slur_homonym_clean`) had an accuracy of just 0.04, and normalized reclaimed slurs (`slur_reclaimed_clean`) achieved an accuracy of 0.12. For the category, `slur_homonym_clean`, the model classified 70% of instances as *Offensive* and 26% as *Hate*, indicating a failure to comprehend the benign sense from context and a tendency to default to the offensive or hateful sense of the word. This pattern underscores the model's strong reliance on lexical cues rather than contextual understanding.

In the case of `slur_reclaimed_clean`, the only normalized reclaimed slur present in HateCheck-XR was "queer," and all corresponding instances were benign (e.g., *"The Q in LGBTQ stands for queer,"* or *"As a queer person myself, I don't understand why people are mad at you."*). Nevertheless, the model misclassified 59% of these as hate speech and 29% as offensive language, likely reflecting its exposure during training on the Davidson dataset, where instances with a mention of "queer" may have been treated as inherently hateful.

Obfuscated spellings Four categories assessed the model's ability to detect hate speech when spellings were altered; all these four only featured hate speech. Single words were modified by omitting a character (accuracy: 0.17), swapping adjacent characters (accuracy: 0.30), using leet speak (accuracy: 0.17), or inserting spaces within a word (accuracy: 0.12). In another category, each test case featured a single pair of words joined without a space (accuracy: 0.37). All these categories yielded low accuracies, demonstrating that the model is highly susceptible to simple spelling

variations and minor textual corruptions. In these cases, instances were more frequently classified as *Clean* than as *Hate*, highlighting the model's limited capacity to recognize hate speech when the text is even slightly obfuscated.

Summary In sum, the results vividly illustrate how a model which looks respectable on the standard test split may crumble when it faces a challenge set, outside the dataset where it was fine-tuned. Moreover, the model relied primarily on obvious lexical cues and struggled with anything that required contextual understanding or interrupted the obvious lexical cues, such as implicit hate, pronoun references, quotations condemning hate, benign uses of "slurs", or minor spelling obfuscations. It was biased toward the offensive class due to the training data imbalance, frequently misclassifying hate speech, and failing when the surface cues were absent.

4.3.3 Trained and Tested on *Hate vs. Non-hate*

In this subsection, I present the results of the model trained to distinguish between hateful and non-hateful instances, and evaluated on the same task. The model was trained on the Davidson dataset and tested on HateCheck-XR. I describe its behavior across different functionality test categories. The earlier observations from the three-class model predicting the three classes are also often found here, and therefore while they are mentioned here, this subsection focuses on observations more unique to this experiment.

Table 4.39 reports the model's accuracy for each functionality test in the second column. The third column presents the results from the previous model, originally trained on three classes, with predictions merged into *Hate* vs. *Non-hate* so that they can be compared with the performance of the *Hate vs. Non-hate* model. The fourth column shows the number of instances in each functionality test.

Overall performance The overall accuracy of the binary model (0.52 ± 0.05) is only slightly higher than pure chance, indicating that distinguishing between *Hate* and *Non-hate* on HateCheck-XR remains a challenging task. The macro F_1 score is similarly low at 0.51 ± 0.05. For comparison, when the ternary predictions of the three-class model were collapsed into the same binary categories, its accuracy dropped to 0.48 ± 0.06 and macro F_1 to 0.47 ± 0.07, which is below the expected performance of random guessing. This shows that training directly on the binary task resulted in modest but consistent improvements (about 4 percentage points both in accuracy and macro F_1) over the three-class system.

The reasons for the generally low scores are consistent with the reasons explained in the earlier part for the ternary class model predicting the three classes, such as difficulty understanding implicit language, that the majority of the Davidson dataset was offensive instances, creative morphological variations for hate speech, the model not having enough world knowledge, sexist and derogatory words against women having been commonly annotated as offensive rather than hateful in the Davidson dataset, poor anaphoric resolution, susceptibility to spelling obfuscations, difficulty understanding quotes and references in counterspeech, and more, as already explained. In short, when the categories of *Offensive* and *Clean* were used as one *Non-hate* category, the boundary between Hate and Non-hate still dealt with the problems.

Functionality	Accuracy	Acc. (3 model→2)	N
counter_quote_clean	0.57±0.12	0.64±0.17	173
counter_ref_clean	0.49±0.17	0.64±0.16	141
derog_dehum_h	0.52±0.11	0.41±0.11	140
derog_impl_h	0.32±0.13	0.24±0.18	140
derog_neg_attrib_h	0.49±0.15	0.38±0.19	140
derog_neg_emote_h	0.57±0.18	0.45±0.23	140
derog_neg_emote_off	1.0±0.0	1.0±0.0	1
ident_neutral_clean	0.59±0.12	0.72±0.19	126
ident_pos_clean	0.66±0.14	0.76±0.12	189
negate_neg_clean	0.48±0.15	0.63±0.23	133
negate_pos_h	0.37±0.1	0.31±0.2	140
phrase_opinion_h	0.48±0.15	0.34±0.18	133
phrase_question_h	0.42±0.1	0.3±0.2	140
profanity_h	0.52±0.13	0.33±0.11	140
profanity_off	1.0±0.0	0.99±0.01	110
ref_subs_clause_h	0.56±0.2	0.44±0.18	140
ref_subs_sent_h	0.6±0.16	0.42±0.17	133
slur_h	0.39±0.06	0.39±0.07	144
slur_homonym_clean	0.66±0.2	0.74±0.19	30
slur_reclaimed_clean	0.33±0.3	0.41±0.3	15
slur_reclaimed_off	0.6±0.13	0.63±0.1	66
spell_char_del_h	0.26±0.12	0.17±0.12	140
spell_char_swap_h	0.42±0.1	0.3±0.14	133
spell_leet_h	0.23±0.07	0.17±0.06	173
spell_space_add_h	0.17±0.06	0.12±0.09	173
spell_space_del_h	0.44±0.12	0.37±0.1	141
target_group_clean	0.82±0.09	0.83±0.1	12
target_group_off	0.84±0.11	0.88±0.09	50
target_indiv_clean	0.91±0.06	0.98±0.02	30
target_indiv_off	0.88±0.14	0.93±0.07	130
target_obj_clean	1.0±0.0	0.99±0.01	64
target_obj_off	1.0±0.0	1.0±0.0	7
threat_dir_h	0.57±0.2	0.44±0.23	133
threat_direct_off	1.0±0.0	1.0±0.0	7
threat_indirect_off	1.0±0.0	1.0±0.0	2
threat_norm_h	0.56±0.15	0.46±0.23	140
threat_norm_off	1.0±0.0	1.0±0.0	6
overall accuracy	0.52 ± 0.05	0.48 ± 0.06	3855
overall macro F_1	0.51 ± 0.05	0.47 ± 0.07	3855

TABLE 4.39: Category-wise accuracy (mean ± SD from five runs) of the model trained and tested under the *Hate vs. Non-hate* setup in the second column, followed by overall accuracy and macro F_1. The third column reports accuracy for the three-class model, with the results re-organized into *Hate vs. Non-hate* for comparison. Column N shows the instance count per category. Trained on Davidson; evaluated on HateCheck-XR.

Perfect scores The model achieved perfect accuracy scores in `threat_direct_off`, `threat_indirect_off`, `threat_norm_off`, and `derog_neg_emote_off`, as well as on `target_obj_clean` and `profanity_off`. Upon inspection, I found none of these categories' instances included a mention of a protected group, though profane words or slurs are sometimes present in the offensive categories. It seems the absence of a protected group identifier strongly cued the model to predict the instances to be *Non-hate*. Consistently, every other `_off` category also received high scores: the category, `target_indiv_off` reached 0.88±0.14 and `slur_reclaimed_off` achieved 0.6±0.13, lower but still noticeably higher than the overall accuracy.

This pattern mirrors the observations from the earlier results where the three-class model predicted three classes, which showed that the model heavily relied on lexical cues; similarly, here, when a protected group mention was missing, the model predicted the instance was non-hateful. A likely reason was the composition of the Davidson training data, where approximately 77% of instances are labeled *Offensive* (i.e., *Non-hate*), and those *Offensive* cases rarely contained protected group identifiers, whereas the *Hate* instances almost always do. As a result, the model learned to associate the absence of protected group identifiers with the non-hate class, and thus defaults to predicting non-hate when explicit hateful target markers are missing.

Higher accuracy scores for detecting hate speech The binary *Hate vs. Non-hate* model achieved higher scores on the hate categories than the ternary model, whose predictions were collapsed into *Hate vs. Non-hate* for comparison (it was also the case when the models were evaluated on the Davidson test set). This is likely because the binary model's training objective directly optimized a single decision boundary aligned with the evaluation task. In contrast, the three-class model also had to distinguish *Offensive* from *Clean*—a task irrelevant to "hate" detection—which introduced an additional boundary that reduced the margin between *Hate* and *Non-hate*. In the Davidson dataset, *Offensive* instances made up about 77% of the data, while *Hate* instances comprised only around 6%. As a result, three-class training was dominated by the majority *Offensive* class, and further burdened by the need to learn the *Offensive–Clean* boundary, leading to more *Hate→Offensive* confusions, which reduced accuracy on hateful instances compared to the binary model trained directly for *Hate vs. Non-hate*. In addition, because the *Hate* and *Offensive* classes share salient lexical cues (e.g., profanity, slurs, hostility), the ternary model more frequently misclassified hateful instances as offensive.

Binary model worse on `_clean` categories The binary model trained to distinguish hate speech from non-hateful instances performed more poorly on the categories about the "clean" language (`_clean` categories), compared to the three-class model's ternary classification results re-organized into *Hate* vs. *Non-hate*. The only exception found was *target_obj_clean* where the binary model scored 1.0±0.0 and the ternary model's re-organized results scored 0.99±0.01, which is a negligible difference. In all other functionality categories involving "clean" cases, the binary model's accuracy was lower by a value between 0.01 and 0.15.

The better scores from the three-class system's re-organized results had a few causes. First, explicit training on the *Clean* class taught the model to pick up cues for benign uses of seemingly-toxic words better (e.g., quoted or referenced hate

speech, normalized reclaimed slurs, benign homonyms of slurs), therefore it decreased *Clean→Hate* misclassifications, which matters for the *Hate vs. Non-hate* classification task. Second, when the three-class predictions were collapsed to *Hate* vs. *Non-hate*, any "clean" items predicted as offensive were counted as correct since both map to *Non-hate*.

Finally, the binary model's representation of *Non-hate* was biased toward the offensive language because the offensive class consisted 77% of the dataset. The true "cleanness" was then less representative in the *Non-hate* class, and "clean" cases were more likely to be misclassified as *Hate* compared to the ternary model.

If the priority was to detect "hate" speech specifically, the binary model performed better; however, the functionality tests showed the vulnerabilities of the model were still in general consistent with the findings from the three-class model's results earlier.

4.3.4 Trained and Tested on *Non-clean vs. Clean*

This section evaluates (i) a RoBERTa-base model fine-tuned directly on the *Non-clean vs. Clean* objective and (ii) a three-class RoBERTa-base model (Hate, Offensive, Clean) whose predictions are collapsed to *Non-clean vs. Clean* for comparison. Both models were trained on the Davidson dataset and tested on HateCheck-XR. Table 4.40 reports accuracy from the functionality tests; the second column shows the binary model's, the third is the results of three-class model's predictions reorganized for the *Non-clean vs. Clean* classification task, and the fourth gives instance counts.

Overall results The *Non-clean vs. Clean* model achieved an overall accuracy of 0.59 ± 0.06 and macro F_1 of 0.53 ± 0.03. Ten functionality test categories reached or exceeded 0.69 in mean accuracy across five runs. Relative to the collapsed three-class model, the binary model performed better by 0.04 in overall accuracy and 0.03 in macro F_1 on this task, as their metrics were 0.55 ± 0.05 and 0.50 ± 0.02 each.

For comparison, in terms of overall accuracy and macro F_1, the model performed better than the *Hate vs. Non-hate*. The binary *Hate vs. Non-hate* model reached 0.52 in accuracy and 0.51 in F_1, while the three-class model collapsed to *Hate vs. Non-hate* did even worse, with 0.48 and 0.47 respectively.

Performance on `_clean` categories As shown in Table 4.40, the *Non-clean vs. Clean* model outperformed the collapsed three-class model on all `_clean` tests (for example, `counter_quote_clean`: 0.53 vs. 0.39; `counter_ref_clean`: 0.46 vs. 0.32; `ident_neutral_clean`: 0.69 vs. 0.57; `negate_neg_clean` 0.65 vs. 0.47 and more), except in `target_indiv_clean` (0.73 vs. 0.81).

With only one decision boundary (*Non-clean vs. Clean*), a benign item has a single failure direction, whereas in the three-class setup it can be mis-directed to either *Hate* or *Offensive* before collapsing, which doubles the routes to a false positive. This helped the binary model avoid flagging "clean" instances of counterspeech, negations, and positive or neutral identity mentions as abusive (*Non-clean*). Moreover, in the training data, *Non-clean* cases were dominated by offensive examples (since the offensive class made up 77% of the Davidson dataset), and features like profanity, slurs, and mentions of protected identities appeared mainly in *Non-clean* and only rarely in *Clean*.

Functionality	Accuracy	Acc. (3 model→2)	N
counter_quote_clean	0.53±0.14	0.39±0.15	173
counter_ref_clean	0.46±0.14	0.32±0.12	141
derog_dehum_h	0.33±0.2	0.51±0.14	140
derog_impl_h	0.21±0.11	0.37±0.17	140
derog_neg_attrib_h	0.4±0.22	0.55±0.15	140
derog_neg_emote_h	0.35±0.11	0.58±0.16	140
derog_neg_emote_off	0.0±0.0	0.0±0.0	1
ident_neutral_clean	0.69±0.1	0.57±0.14	126
ident_pos_clean	0.75±0.11	0.65±0.1	189
negate_neg_clean	0.65±0.12	0.47±0.19	133
negate_pos_h	0.28±0.14	0.44±0.15	140
phrase_opinion_h	0.45±0.15	0.62±0.15	133
phrase_question_h	0.37±0.15	0.54±0.16	140
profanity_h	0.94±0.04	0.97±0.02	140
profanity_off	0.87±0.06	0.91±0.06	110
ref_subs_clause_h	0.42±0.13	0.65±0.13	140
ref_subs_sent_h	0.48±0.15	0.67±0.13	133
slur_h	0.75±0.08	0.75±0.04	144
slur_homonym_clean	0.05±0.02	0.04±0.03	30
slur_reclaimed_clean	0.13±0.0	0.12±0.03	15
slur_reclaimed_off	1.0±0.0	1.0±0.01	66
spell_char_del_h	0.42±0.11	0.49±0.1	140
spell_char_swap_h	0.42±0.1	0.51±0.08	133
spell_leet_h	0.36±0.13	0.4±0.09	173
spell_space_add_h	0.19±0.03	0.24±0.07	173
spell_space_del_h	0.49±0.14	0.57±0.08	141
target_group_clean	0.87±0.05	0.83±0.1	12
target_group_off	0.21±0.07	0.24±0.08	50
target_indiv_clean	0.73±0.23	0.81±0.12	30
target_indiv_off	0.41±0.11	0.39±0.09	130
target_obj_clean	0.97±0.03	0.95±0.03	64
target_obj_off	1.0±0.0	1.0±0.0	7
threat_dir_h	0.32±0.12	0.57±0.17	133
threat_direct_off	0.31±0.06	0.31±0.06	7
threat_indirect_off	0.0±0.0	0.0±0.0	2
threat_norm_h	0.35±0.13	0.57±0.17	140
threat_norm_off	0.4±0.15	0.3±0.18	6
overall accuracy	0.59 ± 0.06	0.55± 0.05	3855
overall macro F_1	0.53 ± 0.03	0.50 ± 0.02	3855

TABLE 4.40: Category-wise accuracy (mean ± SD from five runs) of the model trained and tested under the *Non-clean vs. Clean* setup in the second column. Subsequent rows list overall accuracy and macro F_1. The third column reports accuracy for the three-class model, with the results re-organized into *Non-clean vs. Clean* for comparison. Column N shows the instance count per category. Trained on Davidson; evaluated on HateCheck-XR.

As a result, distinguishing *Clean* from *Non-clean* was easier, since those traits strongly signaled *Non-clean*; the model's tendency to over-rely on such cues did not pose as much of a problem here as in the other experiments, because the experimental setup happened to align well with that over-reliance when it comes to `_clean` categories.

Dependence on profanity and slurs The high scores where explicit profanity or slurs were used (`profanity_h`: 0.94 ± 0.04, `profanity_off`: 0.87 ± 0.06, and `slur_h`: 0.75 ± 0.08) show the model's strong reliance on overt cues.

The accuracy score was also high when the remarks were neutral or positive even when a protected group was mentioned, or if they simply targeted objects, showing a specific sensitivity to profanity and slurs. For example, it was scored quite high in more than just a few categories: `target_obj_off` (1.00 ± 0.00), `target_obj_clean` (0.97 ± 0.03), `target_group_clean` (0.87 ± 0.05). The category, `ident_pos_clean` also scored relatively high (0.75 ± 0.11), with `target_indiv_clean` (0.73 ± 0.23), and `ident_neutral_clean` (0.69 ± 0.10). The model could correctly label many instances that mention protected groups as *Clean* as long as there were no explicit abusive words or, the context was neutral or affirming.

The model failed to understand that homonyms of slurs and normalized reclaimed slurs were benign (`slur_homonym_clean`: 0.05 ± 0.02; `slur_reclaimed_clean`: 0.13 ± 0.0) because of the over-reliance on slurs, and they most likely had not seen them used as homonyms or normalized reclaimed slurs during fine-tuning. In contrast, it achieved a perfect accuracy when it was tested on non-normalized, offensive reclaimed slurs, however, in this case it was because the true labels aligned with the oversensitivity. It indicate that the model was highly sensitive to slurs, and lacked contextual understanding, seeing slurs as an almost-definitive cue for *Non-clean*.

Additionally, more limitations appeared when abuse was implicit or identification of targets was challenging. The test category where dehumanization often occurred without profanity and the category of implicit hate received low scores (`derog_dehum_h`: 0.33 ± 0.20; `derog_impl_h`: 0.21 ± 0.11). While `slur_h` had a good accuracy score of 0.75, it was still relatively lower than that of profanity_h (0.94) or profanity_off (0.87). As described in the earlier part for another experiment, it was largely because the target or hateful words were sometimes less obvious due to morphologically creative blends or compound words, compared to the two profanity categories.

Similarly, even if the instances were direct threats and normative threats, because they lacked profanity or slurs, they also scored poorly (`threat_dir_h`: 0.32 ± 0.12; `threat_dir_off`: 0.31 ± 0.06; `threat_norm_h`: 0.35 ± 0.13; `threat_norm_off`: 0.4 ± 0.15). Additionally, as expected, when they were indirect threats, they scored even more poorly; `threat_indirect_off` achieved 0 in accuracy. (However, note `threat_direct_off`, `threat_indirect_off`, and `threat_norm_off` only had 7, 2, and 6 instances respectively, and therefore the scores should be taken with a grain of salt.) The model's accuracy was below the random-guess baseline (0.5) in the categories with obfuscated spellings, largely because explicit hateful expressions and mentions of protected groups were harder to detect due to the obfuscations. In other words, the model was bad at dealing with obfuscated spellings.

Why the three-class model (collapsed) beats the binary model on hate-involved tests Across functionality categories of hate speech, the collapsed results of the three-class model showed that the three-class model was more sensitive to *weak* abusive signals, yielding higher *Non-clean* recall once collapsed. It can be seen by that the three-class' re-organized predictions scored better in `_h` categories. Because the ternary model had to distinguish the two types of abuse (hate speech and offensive language) during the training, it probably learned lexical patterns of each more successfully.

This showed up as consistent gaps (`derog_dehum_h`: 0.33 vs. 0.51, `derog_impl_h`: 0.21 vs. 0.37, `derog_neg_attrib_h`: 0.40 vs. 0.55, `phrase_opinion_h`: 0.45 vs. 0.62, `phrase_question_h`: 0.37 vs. 0.54, `threat_dir_h`: 0.32 vs. 0.57, and `threat_norm_h`: 0.35 vs. 0.57). Mechanistically: (i) the three-class objective forced the model to learn hate-specific signals distinct from offensive traits; and (ii) when collapsed, an example was counted as *Non-clean* if either *Hate* or *Offensive* beat *Clean*, which was more advantageous compared to the binary model's single *Non-clean vs. Clean* boundary. By contrast, when hate was explicit the gap decreased noticeably (e.g., `profanity_h` 0.94 vs. 0.97; `slur_h` 0.75 vs. 0.75).

It can also explain why the ternary model performed better on all the functionality tests that involved hate speech with obfuscated spellings, and was slightly more robust to obfuscation, picking up on hateful cues and patterns better. The gap was not big, (`spell_char_swap_h`: 0.42 vs. 0.51, `spell_space_del_h`: 0.49 vs. 0.57, `spell_lect_h` 0.36 vs. 0.40), but the binary model which lacked learning about hate speech and offensive language specifically, was less sensitive to those obfuscated hateful expressions.

Summary The binary model was a stronger *Non-clean vs. Clean* detector than the three-class model, but it traded off accuracy for hateful content, especially when hate speech was implicit or hateful words were obfuscated—where the collapsed three-class model did better.

4.3.5 Trained on *Hate vs. Clean*

In this section, I present results of the model trained under the *Hate vs. Clean* setting. The RoBERTa-base model was fine-tuned on the Davidson dataset using only *Hate* and *Clean* instances as all the instances of *Offensive* class were removed. Then it was evaluated on HateCheck-XR.

First, I report its performance on the *Hate vs. Clean* classification task. Next, I present its evaluation results for the *Hate vs. Non-hate* classification task, followed by the *Non-clean vs. Clean* classification task.

1. *Hate vs. Clean* model tested on *Hate vs. Clean*

In this part, results of the *Hate vs. Clean* model's performance on the *Hate vs. Clean* classification task on HateCheck-XR are reported. All functionality tests about *Offensive* instances were excluded during the evaluation. Table 4.41 reports accuracy (mean $\pm$ std) and instance counts per functionality.

Overall performance The model achieved an overall accuracy of 0.59 ± 0.06 and a macro F_1 of 0.54 ± 0.03 across 3,476 test instances, which were the highest accuracy and macro F_1 among all configurations evaluated on HateCheck-XR. This *Hate vs. Clean* model's performance on the *Hate vs. Clean* classification task was the only one

to attain the top values for both metrics simultaneously, though the margins with the next-best settings were small.

Functionality	Accuracy	N
counter_quote_clean	0.32±0.15	173
counter_ref_clean	0.31±0.1	141
derog_dehum_h	0.56±0.13	140
derog_impl_h	0.37±0.13	140
derog_neg_attrib_h	0.65±0.13	140
derog_neg_emote_h	0.74±0.15	140
ident_neutral_clean	0.6±0.14	126
ident_pos_clean	0.64±0.14	189
negate_neg_clean	0.43±0.16	133
negate_pos_h	0.46±0.16	140
phrase_opinion_h	0.77±0.09	133
phrase_question_h	0.62±0.13	140
profanity_h	0.92±0.05	140
ref_subs_clause_h	0.75±0.17	140
ref_subs_sent_h	0.71±0.18	133
slur_h	0.69±0.09	144
slur_homonym_clean	0.41±0.05	30
slur_reclaimed_clean	0.19±0.13	15
spell_char_del_h	0.48±0.11	140
spell_char_swap_h	0.57±0.14	133
spell_leet_h	0.45±0.12	173
spell_space_add_h	0.27±0.08	173
spell_space_del_h	0.63±0.16	141
target_group_clean	0.77±0.07	12
target_indiv_clean	0.85±0.1	30
target_obj_clean	0.99±0.01	64
threat_dir_h	0.76±0.12	133
threat_norm_h	0.78±0.12	140
overall accuracy	0.59±0.06	3476
overall macro F_1	0.54±0.03	3476

TABLE 4.41: Category-wise accuracy (mean ± SD from five runs) of the model trained and tested under the *Hate vs. Clean* setup reported in the second column. Subsequent rows list overall accuracy and macro F_1. Column N shows the instance count per category. Trained on Davidson; evaluated on HateCheck-XR.

Going beyond profanity Compared to the three-class model and the other two binary models, the *Hate vs. Clean* model performed substantially better on subtle hate categories. For `derog_neg_attrib_h` and `derog_neg_emote_h`, where the other models scored in the 0.35–0.57 range, the present model achieved 0.65 ± 0.13 and 0.74 ± 0.15, respectively. Similar gains appeared in the categories, `phrase_opinion_h` and `phrase_question_h`, with accuracies of 0.77 ± 0.09 and 0.62 ± 0.13 compared to 0.30–0.48 for the other models. For `spell_space_del_h`, the model reached 0.63 ± 0.16, while the others scored 0.37–0.49. Finally, for `threat_dir_h` and `threat_norm_h`

which did not include profanity and slurs, the model obtained 0.76 ± 0.12 and 0.78 ± 0.12, surpassing the other model's scores in the 0.32–0.57 range.

These improvements likely reflect the *Hate vs. Clean* model assigning greater relative weight to non-profanity/slur hate cues. In the Davidson dataset, the non-clean portion is heavily dominated by profanity and slurs due to its hateful-lexicon-based data collection, with the *Offensive* class comprising approximately 77% of the corpus. The training configurations that included the *Offensive* class therefore encouraged reliance on surface lexical cues such as profanity and slurs. The resulting class imbalance meant that, in both the three-class and *Non-clean vs. Clean* settings, a substantial share of the gradient signal was devoted to distinguishing *Offensive*, rather than distinguishing *Hate* from the other classes. In the three-class setting in particular, this focus reduced the incentive to learn hate-specific cues.

In the *Hate vs. Non-hate* setting, profanity and slurs were prevalent in the hate-negative class (since most *Non-hate* examples were *Offensive*). Consequently, the model down-weighted profanity and slurs as a signal for hate, leading to lower accuracy on hateful instances containing profanity and slurs due to the conflicting evidence from the negative class. By contrast, in the *Non-clean vs. Clean* setting, profanity and slurs became frequent positive indicators for the *Non-clean* class.

In the *Hate vs. Clean* configuration, removing the *Offensive* class eliminated the conflicting signal present in *Hate vs. Non-hate* and the gradient dominance of *Offensive* in the three-class and *Non-clean vs. Clean* setups. This shift allowed the model to place comparatively greater emphasis on non-profanity hate markers (e.g., targeting protected groups, negative attributions, hostile questions), while profanity still remained a strong positive cue for hate. This training objective aligned closely with the evaluation demands of many hate-focused functionality categories, which required detecting hate without reliance on overt profanity or slurs. Examples include `derog_neg_attrib_h` and `derog_neg_emote_h`, which involve hateful attributes or emotions expressed explicitly, as well as categories with opinionated remarks, rhetorical questions, and hateful threats—all of which convey hate speech with minimal or no use of profanity or slurs.

In terms of the decision boundary, removing *Offensive* likely simplified the decision boundary between *Hate* and *Clean*. In the Davidson dataset, label ambiguity was concentrated near the hate–offensive boundary; in the three-class setting, borderline hateful items were often labeled offensive, which could nudge the model away from hate-specific cues. Under the *Hate vs. Clean* training setting, hate-negatives (*Clean*) contained substantially fewer hate-correlated surface cues (e.g., profanity, explicit threat patterns; neutral identity mentions may still appear), reducing conflicting evidence. By contrast, in the *Hate vs. Non-hate* setting (and for the *Hate* logit within the three-class objective), these cues frequently occurred in the hate-negative class (*Offensive*), affecting the gradients to down-weight them for the *Hate* decision. While having the binary objective instead of three also removed some class competition present in three-class training, the observed higher scores indicate that removing the offensive class, rather than binarization per se, was the primary driver of the stronger generalization to profanity-free hate in the functionality tests. This suggests how the offensive class hindered other models' learning or using other cues, and therefore more-balanced datasets are needed for training.

Quotation and reference The refined hate recognition had drawbacks. Compared to the other models, its performance dropped in `counter_quote_clean` (0.32 ± 0.15)

and `counter_ref_clean` (0.31 ± 0.10), categories where hate speech is quoted or referenced as part of counterspeech. Ironically, it comes from the "improved" recognition of hate speech; as the model used more hate speech cues other than profanity and slurs, it could recognize it better when the quoted or referenced remark was hate speech. The problem was that it could not go beyond the recognition, to understand the instances meant the opposite. This was because counterspeech cases using quotation or reference were rare in the tweet-based Davidon dataset.

Limited improvement on "clean" cases Overall, improvements on "clean" cases were limited. Aside from counterspeech, most clean category accuracies were comparable to other models—some higher, some lower, but without consistent gains. This is partly because, with the offensive class removed, the "clean" category became simpler, excluding borderline cases. As a result, the model learned a less nuanced representation of benign language.

"Clean" in the *Hate vs. Clean* setting encompasses everything that is not hateful, ranging from neutral/positive identity mentions to benign slur homonyms, reclaimed terms, and more. Without offensive examples that challenged simplistic cues, the model tended to over-flag some "clean" items containing identity terms or negative predicates, because it did not learn having them does not necessarily mean the instances are hateful or offensive. As a result, the training sharpened the hate-class representation but left the "clean" class under-represented.

Other difficulties Despite some striking improvements, absolute performance remained modest. Only `profanity_h` (0.92 ± 0.05), `target_indiv_clean` (0.85 ± 0.10), and `target_obj_clean` (0.99 ± 0.01) exceeded 0.77 accuracy.

The model continued to struggle with implicit hate (0.37 ± 0.13), benign slur homonyms (0.41 ± 0.05), and normalized reclaimed slurs (0.19 ± 0.13), though the latter two had small sample sizes (30 and 15 instances respectively). Obfuscated spellings also remained challenging, with accuracies from 0.27 to 0.63. The worst case was `spell_space_add_h` (0.27 ± 0.08), where spaces inserted into words obscured target terms more than letter deletions or swaps.

2. *Hate vs. Clean* model tested on *Hate vs. Non-hate*

The same *Hate vs. Clean* model in the previous subsection was evaluated on its ability to classify HateCheck-XR instances into *Hate* or *Non-hate*. This was done by collapsing all the HateCheck-XR labels so that both the *Offensive* and *Clean* classes were mapped to *Non-hate*, while *Hate* remained unchanged. The model's outputs were left intact; *Hate* predictions were interpreted as hate, and *Offensive* and *Clean* predictions as non-hate. Table 4.42 shows accuracy for each functionality category after this mapping.

Since predictions for *Hate* and *Clean* instances were identical to those in the *Hate vs. Clean* evaluation, this analysis focuses on the categories containing offensive examples. These rows are shaded in the table for easier reference.

Overall scores and performance on offensive categories The *Hate vs. Clean* model attained an overall accuracy of 0.57 ± 0.05 and an overall macro F_1 of 0.54 ± 0.03. On the offensive categories, the model's performance was generally poor. To be specific, 0.35 ± 0.05 for `profanity_off`, 0.0 ± 0.0 for `slur_reclaimed_off`, $0.63 \pm$

0.13 for `target_group_off`, 0.52 ± 0.12 for `target_indiv_off`, and 0.29 ± 0.14 for `target_obj_off`.

Even in small categories ($N < 10$), accuracy remained low except in those with only one or two items. For example, `target_obj_offensive` received 0.29 ± 0.14 in ($N = 7$), `threat_direct_off` ($N = 7$) scored 0.60 ± 0.12, `threat_norm_off` ($N = 6$) scored 0.63 ± 0.18. On the other hand, the categories, `derog_neg_emote_off` and `threat_indirect_off` achieved perfect 1.00 ± 0.00, but they only had 1 and 2 instances respectively.

Dependence on hate speech cues As discussed in Subsection 4.3.5, the *Hate vs. Clean* model learned to use a broader set of hate speech cues beyond profanity and slurs. It did not mean it neglected them; on the contrary, it scored 0.92 ± 0.05 in `profanity_h` during the *Hate vs. Clean* evaluation, for example.

However, when applied to the *Hate vs. Non-hate* task, the same reliance of seeing profanity and slurs as cues for hate speech often led to misclassifications. Low scores in `profanity_off` (0.35 ± 0.05) and `slur_reclaimed_off` (0.0 ± 0.0) indicate that the model frequently labeled these offensive instances as *Hate* due to profanity and slurs. This is logical because the model had not seen any offensive examples, and it also shows the model recognized offensive language's similarities with hate speech (profanity and slurs).

Manual inspection was done on the categories whose size was smaller than 10 instances. `target_obj_off` ($N = 7$) received a low accuracy of 0.29 ± 0.14, and the manual check confirmed that every instance contained profanity (*f**** or *sh***), making the model "misclassify" as hateful. This is consistent with the explanation that profanity strongly biased the model toward a hate prediction. Again, this was not surprising; in training, most profanity and slurs appeared in the *Hate* class because the offensive class was absent.

The tendency meant the accuracy was high in the categories with little or no profanity or slurs. That is, the instances were more likely to be classified correctly as *Non-hate* without profanity or slurs. For example, `target_group_off` (0.63 ± 0.13) and `target_indiv_off` (0.52 ± 0.12) rarely contained profanity (and protected group identifiers as well), so the model was more likely to recognize them as non-hateful. The moderate, yet not perfect scores likely come from other similarities to hate speech in sentence structure or lexical cues, along with a bit of profanity present in the instances.

The pattern was found in the other very small categories as well, though it is difficult to generalize on small size categories. In `derog_neg_emote_off` ($N = 1$) and `threat_indirect_off` ($N = 2$), the absence of profanity and slurs led to perfect scores. In `threat_direct_off` ($N = 7$), during a manual check, it was found two examples contained profanity, lowering accuracy to 0.60 ± 0.12. However, the fact that 2/7 (approximately 0.29) did not directly form the accuracy suggests other similar patterns with hate speech such as sentence structures impacted the model as well.

For `threat_norm_off` ($N = 6$), only one example contained profanity, but it seems strong words like *"kill"* and *"beat up"* in the instances still triggered a similar impact as profanity, therefore creating false positives, yielding 0.63 ± 0.18.

In short, similarities to hate speech, particularly profanity and slurs, but also structural and lexical overlap, were the strongest factors for lowering accuracy on the offensive categories. The model's lack of exposure to non-hateful offensive examples during training made it prone to equating such cues with hate.

Functionality	Accuracy ± std	Instance count
counter_quote_clean	0.32±0.15	173
counter_ref_clean	0.31±0.1	141
derog_dehum_h	0.56±0.13	140
derog_impl_h	0.37±0.13	140
derog_neg_attrib_h	0.65±0.13	140
derog_neg_emote_h	0.74±0.15	140
derog_neg_emote_off	1.0±0.0	1
ident_neutral_clean	0.6±0.14	126
ident_pos_clean	0.64±0.14	189
negate_neg_clean	0.43±0.16	133
negate_pos_h	0.46±0.16	140
phrase_opinion_h	0.77±0.09	133
phrase_question_h	0.62±0.13	140
profanity_h	0.92±0.05	140
profanity_off	0.35±0.05	110
ref_subs_clause_h	0.75±0.17	140
ref_subs_sent_h	0.71±0.18	133
slur_h	0.69±0.09	144
slur_homonym_clean	0.41±0.05	30
slur_reclaimed_clean	0.19±0.13	15
slur_reclaimed_off	0.0±0.0	66
spell_char_del_h	0.48±0.11	140
spell_char_swap_h	0.57±0.14	133
spell_leet_h	0.45±0.12	173
spell_space_add_h	0.27±0.08	173
spell_space_del_h	0.63±0.16	141
target_group_clean	0.77±0.07	12
target_group_off	0.63±0.13	50
target_indiv_clean	0.85±0.1	30
target_indiv_off	0.52±0.12	130
target_obj_clean	0.99±0.01	64
target_obj_off	0.29±0.14	7
threat_dir_h	0.76±0.12	133
threat_direct_off	0.6±0.12	7
threat_indirect_off	1.0±0.0	2
threat_norm_h	0.78±0.12	140
threat_norm_off	0.63±0.18	6
overall accuracy	0.57 ± 0.05	3855
overall macro F_1	0.54 ± 0.03	3855

TABLE 4.42: Category-wise accuracy (mean ± SD from five runs) for a *Hate vs. Clean*–trained model. The second column shows *Hate vs. Non-hate* performance; subsequent rows list overall accuracy and macro F_1. Column N shows the instance count per category. Trained on Davidson; evaluated on HateCheck-XR. Shaded rows denote offensive functionality categories.

Comparison to the *Hate vs. Non-hate* model on offensive categories
When compared to the model trained directly on the *Hate vs. Non-hate* task, the *Hate vs. Clean* model performed markedly worse on offensive categories. For example:

- `profanity_off`: 1.00 vs. 0.35
- `slur_reclaimed_off`: 0.60 vs. 0.00
- `target_group_off`: 0.84 vs. 0.63
- `target_indiv_off`: 0.88 vs. 0.52

Also in small categories ($N < 10$), differences were pronounced: `target_obj_off` (1.00 vs. 0.29), `threat_direct_off` (1.00 vs. 0.60), and `threat_norm_off` (1.00 vs. 0.63). In `derog_neg_emote_off` and `threat_indirect_off`, both models achieved 1.00, but these contained only one and two instances, respectively.

The main reason for this gap is that the *Hate vs. Clean* model never saw offensive examples in training. This reinforced a shortcut: cues frequent in the *Hate* class (slurs, profanity, identity terms) became strong predictors of hate. When evaluated under the *Hate vs. Non-hate* policy, where *Non-hate* included both offensive and "clean" examples, these cues caused the model to misclassify many offensive items as *Hate*, leading to a higher false positive rate and lower accuracy.

By contrast, the *Hate vs. Non-hate* model was trained with a more diverse set of non-hate examples as offensive instances were included, which also means it faced borderline instances as well. It learned richer representations of *Non-hate* and generalized better, which allowed it to down-weight raw offensiveness and focus more on hostility targeted at protected groups. This richer exposure enabled it to generalize better to offensive categories.

In summary, the *Hate vs. Non-hate* model learned to separate *Hate* and *Non-hate* better, highlighting that, with appropriate training, models can reliably distinguish hate speech from merely offensive language.

3. *Hate vs. Clean* model tested on *Non-clean vs. Clean*

This section presents the results of evaluating the same *Hate vs. Clean* model on the *Non-clean vs. Clean* classification task. For this evaluation, HateCheck-XR labels were merged so that both the *Hate* and *Offensive* classes became *Non-clean*, while the original *Clean* class remained unchanged. The model's predictions were not altered; rather, its *Hate* and *Offensive* predictions were interpreted as *Non-clean*, and *Clean* predictions remained *Clean*. Table 4.43 reports the resulting accuracy scores for each functionality category; offensive functionality categories are shaded in the table for ease of reference.

Overall performance While it is not excellent, with the average scores lower than 0.6, they were relatively strong (overall accuracy ≈ 0.59, macro $F_1 \approx 0.53$), on par with the model trained directly on *Non-clean vs. Clean*, macro F_1 only lower by 0.01.

From here on, the *Hate vs. Clean* model's behavior on the task of the *Non-clean vs. Clean* classification and the mechanism behind it is analyzed by comparing it to that of the *Non-clean vs. Clean* model.

Behaviors on hate categories

The *Hate vs. Clean* model performed better in all hate categories on the *Non-clean vs. Clean* classification task, except in `profanity_h` and `slur_h`. In those two categories it achieved scores 0.92 and 0.69 respectively, only lower by 0.02 and 0.06. This

seems to be a small trade-off for the model having learned to use different aspects of hate speech other than profanity and slurs.

Functionality	Accuracy	N
counter_quote_clean	0.32±0.15	173
counter_ref_clean	0.31±0.1	141
derog_dehum_h	0.56±0.13	140
derog_impl_h	0.37±0.13	140
derog_neg_attrib_h	0.65±0.13	140
derog_neg_emote_h	0.74±0.15	140
derog_neg_emote_off	0.0±0.0	1
ident_neutral_clean	0.6±0.14	126
ident_pos_clean	0.64±0.14	189
negate_neg_clean	0.43±0.16	133
negate_pos_h	0.46±0.16	140
phrase_opinion_h	0.77±0.09	133
phrase_question_h	0.62±0.13	140
profanity_h	0.92±0.05	140
profanity_off	0.65±0.05	110
ref_subs_clause_h	0.75±0.17	140
ref_subs_sent_h	0.71±0.18	133
slur_h	0.69±0.09	144
slur_homonym_clean	0.41±0.05	30
slur_reclaimed_clean	0.19±0.13	15
slur_reclaimed_off	1.0±0.0	66
spell_char_del_h	0.48±0.11	140
spell_char_swap_h	0.57±0.14	133
spell_leet_h	0.45±0.12	173
spell_space_add_h	0.27±0.08	173
spell_space_del_h	0.63±0.16	141
target_group_clean	0.77±0.07	12
target_group_off	0.37±0.13	50
target_indiv_clean	0.85±0.1	30
target_indiv_off	0.48±0.12	130
target_obj_clean	0.99±0.01	64
target_obj_off	0.71±0.14	7
threat_dir_h	0.76±0.12	133
threat_direct_off	0.4±0.12	7
threat_indirect_off	0.0±0.0	2
threat_norm_h	0.78±0.12	140
threat_norm_off	0.37±0.18	6
overall accuracy	0.59 ± 0.06	3855
overall macro F_1	0.53 ± 0.03	3855

TABLE 4.43: Category-wise accuracy (mean ± SD from five runs) for a *Hate vs. Clean*–trained model. The second column shows *Non-clean vs. Clean* performance; subsequent rows list overall accuracy and macro F_1. Column N shows the instance count per category. Trained on Davidson; evaluated on HateCheck-XR. Shaded rows denote offensive functionality categories.

In the other 16 hate categories, the *Hate vs. Clean* model performed consistently better across all hate functionalities; the median difference from the improvement was +0.23 and the mean value approximately 0.23. The largest improvements were found in the direct threat category (`threat_dir_h`: +0.44) and the normative threat category (`threat_norm_h`: +0.43). Other categories had considerably performed better as well (`derog_neg_emote_h`: +0.39, `ref_subs_clause_h`: +0.33, and `phrase_opinion_h`: +0.32). The smallest improvements were found in the obfuscated spelling categories `spell_char_del_h` (+0.06) and `spell_space_add_h` (+0.08). The others had moderate gains (0.09-0.25).

This shows how the *Hate vs. Clean* model acquired hate speech cues better, and it considered more than only profanity and slurs. It highlights that learning different cues are crucial for improving hate speech detection across the environment where there are different and difficult hate speech expressions.

The higher scores of the *Hate vs. Clean* model on the functionality tests for hateful cases can be interpreted as the following:

(1) **Objective alignment and gradients** In *Hate vs. Clean* training, the model's optimization focus on separating hate from "cleanness". In *Non-clean vs. Clean*, the *Non-clean* class, *Hate* is present with a much larger *Offensive* portion (around 77% of the Davidson dataset), therefore, gradients were dominated by easy, frequent cues which were profanity and slurs. This diluted learning of hate-specific traits, and impaired detection of hate expressed in different ways.

(2) **Reduced class heterogeneity and label noise** Not having *Offensive* during training yielded a cleaner decision boundary: in comparison to *Non-clean vs. Clean* model, the positive class was more of "hate-only" and the negative class was more of "clean-only." This avoided the noisy hate–offensive border in the Davidson dataset, where borderline items were common, and produced features that generalize to hate without profanity and slurs. It made the model perform better for other categories related to attributions, threats, hateful questions, obfuscated forms.

(3) **Shortcut mitigation** Because *Offensive* was absent in training, profanity and slurs were weaker shortcuts. The model was pushed to rely on other aspects of hate speech more, such as targeted-group markers, negative attributions, and threat semantics, patterns in sentence structures. That explains the big improvements on the categories, `threat_dir_h`, `threat_norm_h`, `derog_neg_emote_h`, `ref_subs_clause_h`, and `phrase_opinion_h`. The small deficits in `profanity_h` and `slur_h` were, as previously mentioned, the trade-off for de-emphasizing crude lexical cues.

(4) **Evaluation mapping** The downside of the same model under the *Hate vs. Non-hate* task was to over-label offensive items as *Hate*, which was wrong in that classification scheme. Then under this *Non-clean vs. Clean* evaluation mapping, those predictions counted as *Non-clean* (i.e., correct), therefore, this behavior no longer harmed performance, but rather improved it. While the model trained for *Non-clean vs. Clean* classification was still stronger on a few offensive categories, the *Hate vs. Clean* model's superior accuracy on hate-specific categories dominates the comparison within the hate subset.

Summary The *Hate vs. Clean* model learned richer hate-specific representations because its training was not overwhelmed by offensive data. That yielded consistent, often large gains on hate functionalities in a *Non-clean vs. Clean* evaluation, at the cost of a very small decrease on profanity-or-slur-heavy hate categories. This

suggests a need for more-balanced datasets where one class does not overwhelm the model in any directions.

Behaviors on clean categories

As mentioned in earlier subsections, the *Hate vs. Clean* model used more various cues of hate speech effectively, but not for the "clean" language very well. Its concept of "cleanness" was more simple and coarse, with less rich representations and it had seen fewer borderline cases without the offensive class during the training, which made its perception of "clean" speech less dimensional as in "what is not hate speech is probably clean."

On the model was assigned to a *Non-clean vs. Clean* task, it performed more correctly on hateful categories, but it misclassified many "clean" cases when they contained hate-like surface cues such as profanity, slurs, and protected identity mentions. Therefore its accuracy on clean functionalities dropped in general, relative to the model that was trained directly on *Non-clean vs. Clean*.

In the counterspeech categories where the instances quoted or referenced hate speech, (`counter_quote_clean`, `counter_ref_clean`), *Hate vs. Clean* model performed ironically worse because it recognized hate speech in these instances better, though it could not understand the instances were talking against hate speech inside them. On the other hand, *Non-clean vs. Clean* model failed to recognize hate speech inside them, which is evidenced by significantly low accuracy scores in most hate categories.

The tendency of taking hate-like words as cues for hate, without considering the context was also found the categories in `slur_homonym_clean`, `slur_reclaimed_clean` and `negate_neg_clean`; the *Hate vs. Clean* model did not understand that the words in the instances of these categories were benign homonym, normalized reclaimed slurs, and negations of hate speech, while it recognized the hate speech part of the instances.

Then in "clean" categories with protected group characteristics mentions, such as `ident_neutral_clean`, `ident_pos_clean`, `target_group_clean`, the *Non-clean vs. Clean* model performed better because it had a better understanding of "clean" language, and *Hate vs. Clean* model probably still took protected group identity mentions as hate speech cues. For those categories, the accuracy scores were 0.69 vs. 0.6, and 0.75 vs. 0.64, and 0.87 vs. 0.77 respectively.

Behaviors on the offensive categories

The performance of the *Hate vs. Clean* model and *Non-clean vs. Clean* model on the offensive categories show differences between them which reflect each model's learned notion of "non-clean" language. The *Hate vs Clean* model, trained to recognize hateful targeting, performed better on targeted offense, whether aimed at individuals or groups (`target_indiv_off`: 0.48 vs. 0.41; `target_group_off`: 0.37 vs. 0.21). By contrast, its performance worse than the other model's when profanity was the main signal (`profanity_off`: 0.65 vs. 0.87; `target_obj_off`: 0.71 vs 1.00, noting N=7), consistent with a decision rule that prioritized targeting cues and whether the targets were people, over profanity as a sufficient indicator of "non-clean" content.

Both models behaved similarly at the extremes: they performed significantly below random chance (0.50) on threats without profanity (`threat_indirect_off`: 0.00 vs 0.00; `threat_norm_off`: 0.37 vs. 0.35 [N=6]) and perfect on reclaimed slurs (`slur_reclaimed_off`: 1.00 vs. 1.00), indicating reliance on lexical cues. Overall, the

Hate vs. Clean model leaned more on signs of who was being targeted (people vs. objects), which explains its weaker performance on object-directed offensive language. During the *Non-clean vs. Clean* model's training, *Non-clean* class was dominated by offensive instances (approximately 77% in the whole Davidson dataset), and therefore it tended to weight profanity more strongly; the heterogeneity and imbalance of its positive class make hate-specific features less necessary to minimize loss.

Taken together, the patterns suggest that *Hate vs. Clean* model considers more various hate-like cues but under-fires on profanity-only offense, while the Non-clean vs. Clean was well-calibrated to profanity but weaker on hate-specific cues. Neither model robustly recognized any threat categories, whether hateful or offensive where profanity was absent, which remains a clear challenge for the goal.

Chapter 5

Discussion

5.1 Answer to the research question & hypothesis

This study investigates the impact of explicitly modeling offensive language as a distinct class on hate speech detection. To this end, language was categorized into mutually exclusive three classes: hateful, offensive, and clean. A series of experiments were conducted in which the offensive class was either included or excluded under various training and evaluation settings. In addition to the ternary classification, certain configurations merged classes to form binary tasks, such as *Hate vs. Non-hate* and *Non-clean vs. Clean*, thereby enabling a more nuanced analysis of how different class distinctions affect model behavior, and how the offensive language impacted the tasks.

While evaluation was performed on the Davidson test split, the need for fine-grained diagnostic insights necessitated the use of a functionality-based test suite as well. For this purpose, HateCheck-XR was developed by re-annotating the original HateCheck to align with the three-class framework, adjusting functionality category names to reflect the new classes, and correcting annotation errors present in the previously mentioned extension. It served not only to improve the diagnostic test set but also to ensure the validity and interpretability of evaluation results, with the underlying ground truth consistent across three classes, even if two of the three classes were merged for experiments.

Turning to the results of the evaluation on these datasets, the evaluation on the test split from the Davidson dataset showed the dominant confusion was *Hate* $\rightarrow$ *Offensive* (Hate speech taken as offensive language), while *Clean* was comparatively easy to separate from *Non-clean*; this is consistent with the dataset's class imbalance (around 6% Hate; 77% Offensive) and with my hypothesis that *Offensive* is closer to *Hate* than to *Clean*.

In contrast, on the evaluation on HateCheck-XR, the same model's dominant confusion came from *Hate* $\rightarrow$ *Clean*, with sizable *Clean* $\rightarrow$ *Hate* confusion as well. This is not a contradiction but a reflection of the kind of test set being used: HateCheck-XR is a diagnostic set with many items that mention words that *seem* hateful but their whole instances are not hateful (e.g. counterspeech, quotation, negation, reclaimed/homonym slurs). A model tuned on the Davidson dataset's simpler cues and hate-scarce environment can mistake them, and also frequently decide more indirectly hateful items are "clean", hence the observed *Hate* $\leftrightarrow$ *Clean* confusions.

Therefore, the statement that "the most difficult part of hate speech detection is separating hateful content from offensive content" may be true for Davidson-dataset-like datasets, which are offensive-heavy and hateful instances are relatively rare. It may not apply to intentionally constructed diagnostic sets such as HateCheck-XR, where the most challenging distinction for the model was *Hate vs. Clean*. In other words, the notion of difficulty appears to be dataset-dependent; it varies with the

evaluation design, types of linguistic phenomena included, and with factors such as class prevalence and decision thresholds, rather than being a fixed, universal property of the task.

Including offensive as a distinct class clarified rather than complicated the task of hate speech detection, and made it easier to understand the model behaviors. The study also supports that training with a separate offensive class (and then collapsing if needed) to capture more borderline hateful content. Moreover, I emphasize the need for diverse challenging data, and for making explicit decisions about the balance between recall and precision, in line with the intended moderation goals.

5.2 Other findings and limitations

In the error analysis per functionality categories of HateCheck-XR, the model showed its limitation that it overly relies on lexical cues that it thought signaled hate speech, such as profanity, slurs, group targeting. It also showed it lacked understanding of the sentence structures and context; e.g. "clean" counterspeech cases quoting or referencing hate speech were classified as hateful because it only recognized the hateful parts. This was most likely because the Davidson dataset on which the models were fine-tuned did not have enough examples of them, considering that the dataset was tweet-based, and only around 6% of it contained hate speech. Identity mentions and targeted-group markers also functioned as superficial cues for the model, and sometimes led to misclassifications of them as hateful in "clean" contexts; however, the model also correctly handled many neutral or positive identity mentions, therefore this underscores the need for richer context modeling rather than simple lexical filtering.

Moreover, the model was not successful at recognizing indirect or implicit hate speech and offensive language, where overt lexical cues were absent, which is still an ongoing challenge in the field, as explained in the Related Work chapter (see 2.2.2).

Additionally, problems in the dataset should be noted. Firstly, the Davidson dataset's annotators tended to annotate sexist or hate speech against women offensive rather than hateful, as noted by Davidson et al. (2017), and this tendency did not align with the present study's definition of hate speech. The authors also noted rarer forms of hate (e.g. hate speech against the Chinese) were neglected, and some mistakes in the annotation.

Also, despite these adjustments and error corrections which made HateCheck-XR, both HateCheck and HateCheck-XR possess limitations. In some cases, instances could reasonably belong to more than one functionality category. For example, instances containing profanity or slurs occasionally appeared in categories not intended to test the model's handling of such language, thereby making it difficult to find which linguistic features attributed to the observed accuracies. In addition, there remains a need to increase the number of offensive examples, particularly in categories with relatively few instances, to create a more balanced and comprehensive test suite.

Lastly, although metrics were averaged across five random seeds, all experiments used the same RoBERTa-base architecture, fixed hyperparameters, and a single-GPU setup. As a result, the results reflect one specific configuration. Comparing RoBERTa-base with RoBERTa-large, DeBERTaV3, or other transformer models with stronger encoders could provide more insight, and might improve the performance.

5.3 Future work

Future studies could explore class-balanced training on the Davidson dataset, using reweighting, cost-sensitive losses, or class-aware sampling. Expanding training data beyond the Davidson dataset or adding a small, representative "challenge" slice with difficult linguistic phenomena may improve cross-dataset robustness, particularly on diagnostic sets like HateCheck-XR and GPT-HateCheck (Jin, Wanner, and Shvets, 2024). Strengthening HateCheck-XR itself by refining functionality categories and using instances that reflect the categories more strictly would make evaluation sharper. On the modeling side, using stronger encoders (e.g., DeBERTaV3_large), parameter-efficient fine-tuning (e.g. Low Rank Adaptation), few-shot target-domain fine-tuning can be considered. These can be combined with robustness/fairness strategies—such as adversarial training and counterfactual augmentation to limit shortcut learning (e.g., over-reliance on identity terms). The code and datasets used and developed for this study are publicly available in the project repository (Kim, 2025) to support replication and further research.

Chapter 6

Conclusion

This thesis explored how treating offensive language as a distinct class, separate from both hateful and "clean" language, affects the model's hate speech detection. Through a series of experiments of multiple classification schemes and two evaluation sets, the results suggest that the difficulty of hate speech detection is not an inherent property of the task itself, but is rather shaped by the data design, class prevalence, and the distribution of linguistic phenomena within the evaluation set. On offensive-heavy datasets like the Davidson dataset, the primary challenge lay in distinguishing hateful content from offensive content; in contrast, the diagnostic suite, HateCheck-XR exposed confusion between the *Hate* class and the *Clean* class, especially in profanity-free or context-dependent cases.

Explicitly modeling offensive language as a separate category did not hinder the task, but instead clarified the sources of model errors and offered a more interpretable framework for both model behavior and evaluation. Including an offensive class enabled more nuanced error analysis and reinforced the importance of capturing borderline or implicit cases of hate speech—instances likely to be missed if a binary *Hate vs. Non-hate* system were enforced. This approach also highlighted that dataset-specific characteristics such as class distribution and types of data (e.g. tweets) can drive model confusion or behavior, and they must be accounted for in both research and real-world usage.

The error analysis revealed key model limitations: over-reliance on lexical cues, insufficient context modeling, and persistent challenges with indirect or implicit hate speech. Dataset annotation inconsistencies, class imbalance, and some overlap between functionality categories also constrained the clarity of evaluation. These limitations suggest that continued progress in hate speech detection requires not just better models or architectures, but also more balanced, diverse, and carefully annotated datasets, as well as diagnostic test suites that reflect the full complexity of real-world language.

Ultimately, this work underscores the need for deliberate, context-sensitive definitions and evaluation practices in hate speech detection research. By making explicit distinctions between hateful, offensive, and "clean" language, and by diagnosing model behavior on challenging test suites, researchers and practitioners can obtain more reliable, actionable insights for content moderation and policy. Future work should build on these findings by improving data diversity, refining annotation standards, and advancing model architectures so that systems may more robustly address the ever-evolving challenges of hateful language online.

Bibliography

Badjatiya, Pinkesh et al. (2017). "Deep Learning for Hate Speech Detection in Tweets". In: *Proceedings of the 26th International Conference on World Wide Web Companion*. WWW '17 Companion. Perth, Australia: International World Wide Web Conferences Steering Committee, 759–760. ISBN: 9781450349147. DOI: 10.1145/3041021.3054223. URL: https://doi.org/10.1145/3041021.3054223.

Burnap, Pete and Matthew L. Williams (2015). "Cyber Hate Speech on Twitter: An Application of Machine Classification and Statistical Modeling for Policy and Decision Making". In: *Policy & Internet* 7.2, pp. 223–242. DOI: https://doi.org/10.1002/poi3.85. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/poi3.85. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/poi3.85.

Caselli, Tommaso et al. (Aug. 2021). "HateBERT: Retraining BERT for Abusive Language Detection in English". In: *Proceedings of the 5th Workshop on Online Abuse and Harms (WOAH 2021)*. Ed. by Aida Mostafazadeh Davani et al. Online: Association for Computational Linguistics, pp. 17–25. DOI: 10.18653/v1/2021.woah-1.3. URL: https://aclanthology.org/2021.woah-1.3/.

Conneau, Alexis et al. (July 2020). "Unsupervised Cross-lingual Representation Learning at Scale". In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Ed. by Dan Jurafsky et al. Online: Association for Computational Linguistics, pp. 8440–8451. DOI: 10.18653/v1/2020.acl-main.747. URL: https://aclanthology.org/2020.acl-main.747/.

Dai, Zihang et al. (July 2019). "Transformer-XL: Attentive Language Models beyond a Fixed-Length Context". In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Ed. by Anna Korhonen, David Traum, and Lluís Màrquez. Florence, Italy: Association for Computational Linguistics, pp. 2978–2988. DOI: 10.18653/v1/P19-1285. URL: https://aclanthology.org/P19-1285/.

Davidson, Thomas et al. (2017). "Automated Hate Speech Detection and the Problem of Offensive Language". In: *Proceedings of the International AAAI Conference on Web and Social Media* 11.1, pp. 512–515. DOI: 10.1609/icwsm.v11i1.14955. URL: https://ojs.aaai.org/index.php/ICWSM/article/view/14955.

Devlin, Jacob et al. (June 2019). "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding". In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Ed. by Jill Burstein, Christy Doran, and Thamar Solorio. Minneapolis, Minnesota: Association for Computational Linguistics, pp. 4171–4186. DOI: 10.18653/v1/N19-1423. URL: https://aclanthology.org/N19-1423/.

Djuric, Nemanja et al. (2015). "Hate Speech Detection with Comment Embeddings". In: *Proceedings of the 24th International Conference on World Wide Web*. WWW '15 Companion. Florence, Italy: Association for Computing Machinery, 29–30. ISBN: 9781450334730. DOI: 10.1145/2740908.2742760. URL: https://doi.org/10.1145/2740908.2742760.

El-Sayed, Ahmed and Omar Nasr (Mar. 2024). "AAST-NLP at Multimodal Hate Speech Event Detection 2024 : A Multimodal Approach for Classification of Text-Embedded Images Based on CLIP and BERT-Based Models." In: *Proceedings of the 7th Workshop on Challenges and Applications of Automated Extraction of Socio-political Events from Text (CASE 2024)*. Ed. by Ali Hürriyetoğlu et al. St. Julians, Malta: Association for Computational Linguistics, pp. 139–144. URL: https://aclanthology.org/2024.case-1.19/.

ElSherief, Mai et al. (Nov. 2021). "Latent Hatred: A Benchmark for Understanding Implicit Hate Speech". In: *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. Ed. by Marie-Francine Moens et al. Online and Punta Cana, Dominican Republic: Association for Computational Linguistics, pp. 345–363. DOI: 10.18653/v1/2021.emnlp-main.29. URL: https://aclanthology.org/2021.emnlp-main.29/.

Fortuna, Paula and Sérgio Nunes (July 2018). "A Survey on Automatic Detection of Hate Speech in Text". In: *ACM Comput. Surv.* 51.4. ISSN: 0360-0300. DOI: 10.1145/3232676. URL: https://doi.org/10.1145/3232676.

Fortuna, Paula, Juan Soler, and Leo Wanner (May 2020). "Toxic, Hateful, Offensive or Abusive? What Are We Really Classifying? An Empirical Analysis of Hate Speech Datasets". eng. In: *Proceedings of the Twelfth Language Resources and Evaluation Conference*. Ed. by Nicoletta Calzolari et al. Marseille, France: European Language Resources Association, pp. 6786–6794. ISBN: 979-10-95546-34-4. URL: https://aclanthology.org/2020.lrec-1.838/.

Galinsky, Adam D. et al. (2013). "The Reappropriation of Stigmatizing Labels: The Reciprocal Relationship Between Power and Self-Labeling". In: *Psychological Science* 24.10, pp. 2020–2029. DOI: 10.1177/0956797613482943. URL: https://doi.org/10.1177/0956797613482943.

Gao, Lei, Alexis Kuppersmith, and Ruihong Huang (Nov. 2017). "Recognizing Explicit and Implicit Hate Speech Using a Weakly Supervised Two-path Bootstrapping Approach". In: *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. Ed. by Greg Kondrak and Taro Watanabe. Taipei, Taiwan: Asian Federation of Natural Language Processing, pp. 774–782. URL: https://aclanthology.org/I17-1078/.

Gupta, Soumyajit, Maria De-Arteaga, and Matthew Lease (2025). *Fairly Accurate: Fairness-aware Multi-group Target Detection in Online Discussion*. arXiv preprint, version 2 (25 Jun 2025). DOI: 10.48550/ARXIV.2407.11933. arXiv: 2407.11933 [cs.LG]. URL: https://arxiv.org/abs/2407.11933.

He, Pengcheng et al. (2021). "DEBERTA: DECODING-ENHANCED BERT WITH DISENTANGLED ATTENTION". In: *International Conference on Learning Representations*. URL: https://openreview.net/forum?id=XPZIaotutsD.

Jin, Yiping, Leo Wanner, and Alexander Shvets (May 2024). "GPT-HateCheck: Can LLMs Write Better Functional Tests for Hate Speech Detection?" In: *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*. Ed. by Nicoletta Calzolari et al. Torino, Italia: ELRA and ICCL, pp. 7867–7885. URL: https://aclanthology.org/2024.lrec-main.694/.

Khurana, Urja, Eric Nalisnick, and Antske Fokkens (Jan. 2025). "DefVerify: Do Hate Speech Models Reflect Their Dataset's Definition?" In: *Proceedings of the 31st International Conference on Computational Linguistics*. Ed. by Owen Rambow et al. Abu Dhabi, UAE: Association for Computational Linguistics, pp. 4341–4358. URL: https://aclanthology.org/2025.coling-main.293/.

Khurana, Urja et al. (July 2022). "Hate Speech Criteria: A Modular Approach to Task-Specific Hate Speech Definitions". In: *Proceedings of the Sixth Workshop on Online Abuse and Harms (WOAH)*. Ed. by Kanika Narang et al. Seattle, Washington (Hybrid): Association for Computational Linguistics, pp. 176–191. DOI: 10.18653/v1/2022.woah-1.17. URL: https://aclanthology.org/2022.woah-1.17/.

Kim, Areumbyeol (2025). *hatespeech-offensive*. https://github.com/areumb/hatespeech-offensive.

Kumar, Gokul Karthik and Karthik Nandakumar (Dec. 2022). "Hate-CLIPper: Multimodal Hateful Meme Classification based on Cross-modal Interaction of CLIP Features". In: *Proceedings of the Second Workshop on NLP for Positive Impact (NLP4PI)*. Ed. by Laura Biester et al. Abu Dhabi, United Arab Emirates (Hybrid): Association for Computational Linguistics, pp. 171–183. DOI: 10.18653/v1/2022.nlp4pi-1.20. URL: https://aclanthology.org/2022.nlp4pi-1.20/.

Kwok, Irene and Yuzhou Wang (2013). "Locate the Hate: Detecting Tweets against Blacks". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 27. 1. AAAI Press, pp. 1621–1622. DOI: 10.1609/aaai.v27i1.8539. URL: https://doi.org/10.1609/aaai.v27i1.8539.

Lhoest, Quentin et al. (Nov. 2021). "Datasets: A Community Library for Natural Language Processing". In: *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*. Ed. by Heike Adel and Shuming Shi. Online and Punta Cana, Dominican Republic: Association for Computational Linguistics, pp. 175–184. DOI: 10.18653/v1/2021.emnlp-demo.21. URL: https://aclanthology.org/2021.emnlp-demo.21/.

Liu, Yinhan et al. (2019). *RoBERTa: A Robustly Optimized BERT Pretraining Approach*. https://arxiv.org/abs/1907.11692. arXiv:1907.11692, doi:10.48550/arXiv.1907.11692.

Masood, Muhammad Arslan et al. (2025). "Balancing Imbalanced Toxicity Models: Using MolBERT with Focal Loss". In: *AI in Drug Discovery*. Ed. by Djork-Arné Clevert et al. Cham: Springer Nature Switzerland, pp. 82–97. ISBN: 978-3-031-72381-0. DOI: 10.1007/978-3-031-72381-0_8.

Pitsilis, Georgios K., Heri Ramampiaro, and Helge Langseth (Dec. 2018). "Effective Hate-Speech Detection in Twitter Data Using Recurrent Neural Networks". In: *Applied Intelligence* 48.12, pp. 4730–4742. DOI: 10.1007/s10489-018-1242-y. URL: https://link.springer.com/article/10.1007/s10489-018-1242-y.

Poletto, Fabio et al. (2021). "Resources and benchmark corpora for hate speech detection: a systematic review". In: *Language Resources and Evaluation* 55.2, pp. 477–523. DOI: 10.1007/s10579-020-09502-8. URL: https://doi.org/10.1007/s10579-020-09502-8.

Röttger, Paul et al. (Aug. 2021). "HateCheck: Functional Tests for Hate Speech Detection Models". In: *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. Ed. by Chengqing Zong et al. Online: Association for Computational Linguistics, pp. 41–58. DOI: 10.18653/v1/2021.acl-long.4. URL: https://aclanthology.org/2021.acl-long.4/.

Saleh, Hind, Areej Alhothali, and Kawthar Moria (2023). "Detection of Hate Speech using BERT and Hate Speech Word Embedding with Deep Model". In: *Applied Artificial Intelligence* 37.1, p. 2166719. DOI: 10.1080/08839514.2023.2166719. URL: https://doi.org/10.1080/08839514.2023.2166719.

Sigurbergsson, Gudbjartur Ingi and Leon Derczynski (May 2020). "Offensive Language and Hate Speech Detection for Danish". eng. In: *Proceedings of the Twelfth*

Language Resources and Evaluation Conference. Ed. by Nicoletta Calzolari et al. Marseille, France: European Language Resources Association, pp. 3498–3508. ISBN: 979-10-95546-34-4. URL: https://aclanthology.org/2020.lrec-1.430/.

Vaswani, Ashish et al. (2017). "Attention is All you Need". In: *Advances in Neural Information Processing Systems*. Ed. by I. Guyon et al. Vol. 30. Curran Associates, Inc. URL: https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

Warner, William and Julia Hirschberg (June 2012). "Detecting Hate Speech on the World Wide Web". In: *Proceedings of the Second Workshop on Language in Social Media*. Ed. by Sara Owsley Sood, Meenakshi Nagarajan, and Michael Gamon. Montréal, Canada: Association for Computational Linguistics, pp. 19–26. URL: https://aclanthology.org/W12-2103/.

Waseem, Zeerak and Dirk Hovy (June 2016). "Hateful Symbols or Hateful People? Predictive Features for Hate Speech Detection on Twitter". In: *Proceedings of the NAACL Student Research Workshop*. Ed. by Jacob Andreas, Eunsol Choi, and Angeliki Lazaridou. San Diego, California: Association for Computational Linguistics, pp. 88–93. DOI: 10.18653/v1/N16-2013. URL: https://aclanthology.org/N16-2013/.

Waseem, Zeerak et al. (Aug. 2017). "Understanding Abuse: A Typology of Abusive Language Detection Subtasks". In: *Proceedings of the First Workshop on Abusive Language Online*. Ed. by Zeerak Waseem et al. Vancouver, BC, Canada: Association for Computational Linguistics, pp. 78–84. DOI: 10.18653/v1/W17-3012. URL: https://aclanthology.org/W17-3012/.

Wiegand, Michael, Josef Ruppenhofer, and Elisabeth Eder (June 2021). "Implicitly Abusive Language – What does it actually look like and why are we not getting there?" In: *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Ed. by Kristina Toutanova et al. Online: Association for Computational Linguistics, pp. 576–587. DOI: 10.18653/v1/2021.naacl-main.48. URL: https://aclanthology.org/2021.naacl-main.48/.

Wolf, Thomas et al. (Oct. 2020). "Transformers: State-of-the-Art Natural Language Processing". In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*. Ed. by Qun Liu and David Schlangen. Online: Association for Computational Linguistics, pp. 38–45. DOI: 10.18653/v1/2020.emnlp-demos.6. URL: https://aclanthology.org/2020.emnlp-demos.6/.

Xhonneux, Sophie et al. (2024). "Efficient Adversarial Training in LLMs with Continuous Attacks". In: *Advances in Neural Information Processing Systems*. Ed. by A. Globerson et al. Vol. 37. Curran Associates, Inc., pp. 1502–1530. URL: https://proceedings.neurips.cc/paper_files/paper/2024/file/0302fb83c62991efbccf0a003e4f5a92-Paper-Conference.pdf.

Xu, Zhi and Sencun Zhu (July 2010). "Filtering Offensive Language in Online Communities Using Grammatical Relations". In: *Proceedings of the 7th Annual Collaboration, Electronic Messaging, Anti-Abuse and Spam Conference (CEAS 2010)*. Redmond, Washington, USA, pp. 1–10. URL: https://citeseerx.ist.psu.edu/document?doi=bd7fa636a644ef1889c0ce5efe2036ff31cff320.

Yang, Zhilin et al. (2019). "XLNet: Generalized Autoregressive Pretraining for Language Understanding". In: *Advances in Neural Information Processing Systems*. Ed. by H. Wallach et al. Vol. 32. Curran Associates, Inc. URL: https://proceedings.neurips.cc/paper_files/paper/2019/file/dc6a7e655d7e5840e66733e9ee67cc69-Paper.pdf.

Yarowsky, David (1994). "Decision lists for lexical ambiguity resolution: application to accent restoration in Spanish and French". In: *Proceedings of the 32nd Annual Meeting on Association for Computational Linguistics*. ACL '94. Las Cruces, New Mexico: Association for Computational Linguistics, 88–95. DOI: 10.3115/981732.981745. URL: https://doi.org/10.3115/981732.981745.

Yuan, Lanqin and Marian-Andrei Rizoiu (2025). "Generalizing Hate Speech Detection Using Multi-Task Learning: A Case Study of Political Public Figures". In: *Computer Speech & Language* 89, p. 101690. ISSN: 0885-2308. DOI: https://doi.org/10.1016/j.csl.2024.101690. URL: https://www.sciencedirect.com/science/article/pii/S0885230824000731.

Zampieri, Marcos et al. (June 2019). "Predicting the Type and Target of Offensive Posts in Social Media". In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Ed. by Jill Burstein, Christy Doran, and Thamar Solorio. Minneapolis, Minnesota: Association for Computational Linguistics, pp. 1415–1420. DOI: 10.18653/v1/N19-1144. URL: https://aclanthology.org/N19-1144/.

Zhang, Zhehao, Jiaao Chen, and Diyi Yang (Dec. 2023). "Mitigating Biases in Hate Speech Detection from A Causal Perspective". In: *Findings of the Association for Computational Linguistics: EMNLP 2023*. Ed. by Houda Bouamor, Juan Pino, and Kalika Bali. Singapore: Association for Computational Linguistics, pp. 6610–6625. DOI: 10.18653/v1/2023.findings-emnlp.440. URL: https://aclanthology.org/2023.findings-emnlp.440/.

Zsisku, Eszter, Arkaitz Zubiaga, and Haim Dubossarsky (2024). "Hate Speech Detection and Reclaimed Language: Mitigating False Positives and Compounded Discrimination". In: *Proceedings of the 16th ACM Web Science Conference*. WEBSCI '24. Stuttgart, Germany: Association for Computing Machinery, 241–249. ISBN: 9798400703348. DOI: 10.1145/3614419.3644025. URL: https://doi.org/10.1145/3614419.3644025.