Quantifying the Role of Contextual Signals for Modelling Hateful Text

A THESIS

SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF

DOCTOR OF PHILOSOPHY

BY

SARAH MASUD (PHD19020)

Computer Science and Engineering

INDRAPRASTHA INSTITUTE OF INFORMATION TECHNOLOGY DELHI

NEW DELHI- 110020

September 4, 2025

THESIS COMMITTEE

Supervisors

- Tanmoy Chakraborty, Associate Professor, Indian Institute of Technology, Delhi
- Vikram Goyal, Professor, Indraprastha Institute of Information Technology, Delhi

Internal Committee

- Md. Shad Akhtar, Assistant Professor, Indraprastha Institute of Information Technology, Delhi
- Raghava Mutharaju, Associate Professor, Indraprastha Institute of Information Technology, Delhi

PhD Comprehensive External Examiner

- Srinath Srinivasa, Professor, International Institute of Information Technology, Bangalore

PhD Thesis External Examiners

- Anupam Joshi, Professor, University of Maryland, Baltimore County
- Steven Schockaert, Professor, Cardiff University
- Sushmita Mitra, Professor, Indian Statistical Institute, Kolkata

THESIS CERTIFICATE

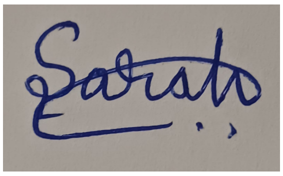
This is to certify that the thesis titled **Quantifying the Role of Contextual Signals in Modelling Hateful Text**, submitted by **Sarah Masud (PHD19020)**, to the Indraprastha Institute of Information Technology Delhi, for the award of the degree of **Doctor of Philosophy**, is a bona fide record of the research work done by her under our supervision. In my opinion, the thesis has reached the standard, fulfilling the requirements of the regulations relating to the degree. The contents of this thesis, in full or in parts, have not been submitted to any other institution or University for the award of any degree or diploma.

Dr. Tanmoy Chakraborty
Thesis Supervisor
Associate Professor
Dept. of Electrical Engineering
IIT Delhi, 110016

Dr. Vikram Goyal
Thesis Supervisor
Professor
Dept. of Computer Science & Engineering
IIIT Delhi, 110020

CANDIDATE’S DECLARATION

The author hereby declares that the work presented in the thesis titled **Quantifying the Role of Contextual Signals in Modelling Hateful Text**, submitted as partial fulfilment for the award of the degree of Doctor of Philosophy to the IIIT-Delhi, is an original research work carried out under the joint supervision of Dr. Tanmoy Chakraborty and Dr. Vikram Goyal. The results presented in this thesis have not been submitted in part or whole to any other university or institute for the award of any degree/diploma.



Sarah Masud (PHD19020)
Candidate
Google PhD 2023-2024
PMRF with WiproAI 2020-2023
Department of Computer Science
IITT Delhi, 110020

"There's this piece of wall in Hiroshima that was completely burnt black by the radiation. But on the front step, a person who was sitting there blocked the rays from hitting the stone. The only thing left now is a permanent shadow of positive light."

-Sarah Kay; Hiroshima

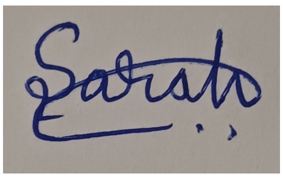
ACKNOWLEDGEMENTS

This thesis would not have been possible without the constant guidance, support, and nudging of my supervisors, Dr Tanmoy Chakraborty and Dr Vikram Goyal. I am incredibly grateful to Dr Tanmoy for having faith in my ability to grow and thrive as a researcher and, most importantly, for introducing me to the world of social computing. It was his conviction and support towards his students that helped me obtain both of my fellowships. Both Dr Tanmoy and Dr Vikram have been generous in providing organisational support to navigate the PhD administration, including the ORF fellowship. This thesis would not be possible without the financial and industrial support of Wipro AI. Thanks to Amitava Das and Vinutha BN for conceptualising and leading the collaboration.

During my PhD, I have had the benefit of receiving mentorship from Dr Shad Akhtar and Dr Subhabrata Dutta. In the last 5 years, interactions with my supervisors and mentors have helped shape my worldviews about varying aspects of life. I want to thank Dr Shad and Dr Raghava for serving on my internal annual committee and providing regular feedback, as well as Dr Srinath Srinivasa for serving on my comprehensive committee. Towards the end of the thesis, I would like to thank my external committee, Dr Anupam Joshi, Dr Steven Schockaert and Dr Sushmita Mitra. This thesis would not be possible without my excellent collaborators, Ashutosh, Aflah, Sahajpreet, Dr Viktor, Dr Alexander, Neemesh, Atharva, Manjot, Sakshi, Chhavi, Dhruv, Vasu, Tanmay, and Tharun. I extend thanks to Dr Jurgen, Vishwajeet, Manav, Aflah and Neemesh for tinkering with ideas that eventually led to new project avenues. I would also like to thank the multiple anonymous conference and journal reviewers who provided timely and critical feedback that helped improve the research papers included in this thesis. Special thanks to Microsoft India, ACM-India, and ACM SIG travel grants. I also want to thank Neha and Vishal from FICCI, Rishabh and Divy from Google India, and Imran, Sarika and Raju at IIIT-Delhi for coordinating my PMRDF (DST-SERB) and Google fellowships. The whole admin and support staff at IIIT-Delhi need to be thanked for keeping the workplace functioning and for ensuring our physical safety within the campus. The role of safe and positive nudging cannot be overstated, and

here I want to thank people from my pre-PhD life who have inspired and supported my journey in STEM. My high school science teachers, Asthana Maam and David Maam; my undergraduate supervisor, Dr. Tanvir Ahmad; and my colleagues at Red Hat. They all let me be a curious kid, take up projects and tasks beyond my skill set and allow me to fail and learn. Here, I want to thank my family. My parents, Nikhat Shafiq and Masud Akhtar, along with my grandparents, aunts, uncles, and first cousins, have, in numerous ways, contributed to my development (and continue to do so), to which I will always be in debt. This is also a shout-out to my cool brother Umar Masud, who is both my source of competition and inspiration. Your dedication, work ethic, and understanding of the world motivate me to be a better person. Thank you for being my guinea pig for data annotation exercises.

Talking about family, I want to thank my friends who have given me the space to exist in my being. Adhbhuti and Zishan, even in your death, you continue to inspire this world for the better. While a PhD is siloed by its design, I never believed I could make so many friends along the way! This journey would not have been half as much fun without my labmates Shivani, Megha, Aseem and Yash. I also want to thank all my peers and research colleagues, especially Venkatesh, Anam, Karan, Kainat, Gunjan, Piyush, and Anupriya, for engaging in discussions with me and guiding me with fellowships. Lastly, thank you to my favourite people on earth for being my ride-or-die – Abhishek, Neha, Sultana, Antriksh, Samar, Zeya and Farheen. You guys literally allowed me to live under a rock without taking my lack of social engagement as a rejection of our friendship. Your journey inspires me to stay grounded and curious. You keep my faith in God and humanity alive. Writing this thesis, I am appreciative of everyone who has, directly and indirectly, contributed towards this journey. I stand in awe of all people whose stories I had the privilege of learning about through books as well as online mediums. I hope to pay forward what I keep learning from the world around me about the ways of life, of which research is but one aspect.



- Sarah Masud

ABSTRACT

KEYWORDS: Hate Speech, Multilingualism, Textual Modality, Contextual Priming, Social Media

Despite our best efforts, tackling hate speech remains an elusive issue for researchers and practitioners alike. What can be considered hateful is subject to context, time, geography, and culture. This poses a challenge in defining standard benchmarks and modelling techniques to combat hate. However, what underpins hate is universally accepted as the intent of dehumanising and biasing against a historically vulnerable group. Unfortunately, determining both intent and power dynamics in an online setting is formidable; further, the influence of the human evaluator's lived experiences creates a gap in the human and computational understanding of hatefulness.

By examining the role of external priming via contextual signals, we aim to bridge this information gap and improve the human-computer alignment for analysing and monitoring hateful content on the Web.

Through a series of five datasets and model pairs, the thesis empirically establishes the efficacy of contextual signals in modelling hate speech-related tasks. The compelling use of contextual signals gets further solidified as our findings apply to any pipeline from feature-engineered logistic regressor to zero-shot prompted large language models. However, we caution against using a one-size-fits-all setup by quantifying the toxic connotations and scalability challenges of certain signals. To this end, the thesis outlines strategies for deployable, human-centric tools for reactive and proactive moderation paradigms, focusing on the multilingual and implicit nature of hate.

LIST OF PAPERS

Journals

J1. Tanmoy Chakraborty, **Sarah Masud**: The Promethean Dilemma of AI at the Intersection of Hallucination and Creativity. **Communication of the ACM, 2024**. 67(10): 26-28 (2024).

J2. **Sarah Masud**, Ashutosh Bajpai, Tanmoy Chakraborty: Focal Inferential Infusion Coupled with Tractable Density Discrimination for Implicit Hate Speech Detection. **Natural Language Engineering (NLE), 2024.** 1–27.

J3. Tanmay Garg, **Sarah Masud**, Tharun Suresh, Tanmoy Chakraborty: Handling Bias in Toxic Speech Detection: A Survey. **ACM Computing Surveys, 2023.** 55(13s): 264:1-264:32.

J4. Tanmoy Chakraborty, **Sarah Masud**: Judging the creative prowess of AI. **Nature Machine Intelligence, 2023.** 5(6): 558.

J5. **Sarah Masud**, Tanmoy Chakraborty: Political mud slandering and power dynamics during Indian assembly elections. **Social Network Analysis and Mining (SNAM), 2023.** Min. 13(1): 108 (2023).

J6. Dhruv Sehnan, Vasu Goel, **Sarah Masud**, Chhavi Jain, Vikram Goyal, Tanmoy Chakraborty: DiVA: A Scalable, Interactive and Customizable Visual Analytics Platform for Information Diffusion on Large Networks. **ACM Transactions on Knowledge Discovery from Data (ACM TKDD), 2023.** 17(4): 47:1-47:33.

J7. Tanmoy Chakraborty, **Sarah Masud**: Nipping in the bud: detection, diffusion and mitigation of hate speech on social media. **ACM SIGWEB Newsletter, 2022.** 3:1-3:9 (2022).

Conferences

C1. **Sarah Masud**, Sahajpreet Singh, Viktor Hangya, Alexander Fraser, Tanmoy Chakraborty: Hate Personified: Investigating the role of LLMs in content moderation. **Proceedings of the Conference on Empirical Methods in Natural**

Language Processing (EMNLP), 2024. 15847-15863.

C2. Neemesh Yadav, **Sarah Masud**, Vikram Goyal, Md. Shad Akhtar, Tanmoy Chakraborty: Tox-BART: Leveraging Toxicity Attributes for Explanation Generation of Implicit Hate Speech. **Findings of the Association for Computational Linguistics (ACL), 2024.** 13967-13983

C3. **Sarah Masud**, Mohammad Aflah Khan, Vikram Goyal, Md. Shad Akhtar, Tanmoy Chakraborty: Probing Critical Learning Dynamics of PLMs for Hate Speech Detection. **Findings of the Association for Computational Linguistics (EACL), 2024.** 826-845.

C4. Atharva Kulkarni, **Sarah Masud**, Vikram Goyal, Tanmoy Chakraborty: Revisiting Hate Speech Benchmarks: From Data Curation to System Deployment. **Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (SIGKDD), 2023**. 4333-4345.

C5. **Sarah Masud**, Manjot Bedi, Mohammad Aflah Khan, Md. Shad Akhtar, Tanmoy Chakraborty: Proactively Reducing the Hate Intensity of Online Posts via Hate Speech Normalization. **Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (SIGKDD), 2022**. 3524-3534

C6. **Sarah Masud**, Subhabrata Dutta, Sakshi Makkar, Chhavi Jain, Vikram Goyal, Amitava Das, Tanmoy Chakraborty: Hate is the New Infodemic: A Topic-aware Modeling of Hate Speech Diffusion on Twitter. **IEEE 37th International Conference on Data Engineering (ICDE), 2021**. 504-515.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

⚠ DISCLAIMER ⚠

The thesis employs explicit examples of hateful and offensive speech solely for the purpose of scientific discussion. The input samples are taken from social media posts and do not reflect the views of the author.

Part I

Introduction

CHAPTER 1

Thesis Introduction

During the French Revolution in 1789, the king first learned of the Storming of the Bastille only the following day. Fast forward to the Arab Spring of 2011, when online social networks (OSNs) were put to productive use in mobilising protests and garnering international support in real time. The emergence of OSNs has truly opened up possibilities that never existed before.
- Sarah Masud; PhD Thesis

1.1 Background

Online social networks, or OSNs, can be described as platforms that enable a network of people to digitally 'socialise' by sharing and exchanging information on the platform. Today, anyone with access to OSNs can learn, share, and participate in matters that concern them. However, with fluid access to information comes the problem of accountability, factuality, and anonymity. The online disinhibition effect (Suler, 2004), fueled by anonymity, has led to an increase in anti-social and harmful behaviour, such as cyberbullying, spamming, and spreading of ill-informed and hateful messages.

In the contemporary world, hate-mongering against various groups has led to genocide (Tsesis, 2002) from Germany (1941-1945) to Rwanda (1994) to modern-day Rohingya (2017-ongoing). On the other hand, systemic discrimination and stereotyping via identity-based profiling (Nadal *et al.*, 2021) negatively impact those at the intersection of race, sexuality, religion, ethnicity, political ideology, etc. With our lives becoming intertwined with digital spaces, we now observe the same institutionalised hatefulness being expressed online. A seemingly benign, unintended, yet uninformed content about a person or a group can, over time, accumulate into hatefulness (Dahiya *et al.*, 2021) and escalate into bias-motivated violence. Ample reports[1] and surveys point towards an increase in online hate speech[2]. The rampant increase in online hate speech gets amplified due to the network effect (Munn, 2020) and the negligible cost of online communication. A higher desensitisation of the public towards online hate speech consequently contributes to a higher level of physical violence and hate crimes being perpetuated in the real world. On the other hand, an increase in instances of hate crimes leads to a rise in hate speech in digital spaces (Lupu *et al.*, 2023) toward the vulnerable groups. Such was the case of George Floyd's murder and #BlackLivesMatter. Different countries have varying legal provisions for accommodating free speech vs hate speech. The First Amendment in the United States of America (USA), Article 19 of the Indian Constitution, and Article 5 of the German Basic Law are examples of constitutional rights to freedom of expression, of which speech is an integral part. In India, Article 19 further states that restrictions on freedom of expression will become applicable if the speech threatens the peace and integrity of the society (such as Section 295A against

[1] https://www.cfr.org/backgrounder/hate-speech-social-media-global-comparisons
[2] https://fra.europa.eu/en/publication/2023/online-content-moderation

religious hatred). Meanwhile, in Germany, holocaust denialism is a criminal offence. However, there is no one legal definition of hate speech that is applicable to all countries.

In absence of a universally accepted definition of hate speech, the United Nations defines it as[3] "*any communication in speech, writing or behaviour, that attacks or uses pejorative or discriminatory language with reference to a person or a group on the basis of who they are, in other words, based on their religion, ethnicity, nationality, race, colour, descent, gender or other identity factors.*"

1.1.1 Terminology

Identity factors in the above definition refer to any attribute that can be associated with a person or a group, allowing them to be differentiated. Some of these factors, as described below, need not be overt:

- **Demographic attributes.** It refers to the statistical characteristics of a population defined by attributes such as age, race, ethnicity, caste, gender, income level, education level, etc. It should be noted that the role of demographic attributes can vary across regions. For example, while 'race' is a defining characteristic in the USA, in India, it is replaced by the notion of 'caste'.
- **Psychographic attributes.** It refers to the characteristics of a population defined by their belief systems, behaviours, interests, hobbies, social class, lifestyles, etc. It can range from religious or political outlook to taste in music. Psychographic attributes can play a role in defining the cognitive characteristics of the population. For example, a population's outlook on homosexuality is driven more by religious and political attributes than, say, by race or caste.
- **Stereotype.** It refers to a negatively intended generalisation of a particular characteristic, behaviour, or trait associated with a demographic or a psychographic group of people. Even seemingly neutral or positive stereotypes are harmful (Wong and Halgin, 2006). For example, saying 'Asians are good at Maths' is loaded with the notion that 'Asians' are not good at anything else!
- **Vulnerable/Target groups.** It refers to people who have historically in the real world been oppressed or abused based on demographic or psychographic attributes (Kulkarni *et al.*, 2023). Despite the legal prevalence of 'protected class' in some countries, such as the USA, the concept of vulnerable or target groups in hate speech literature is contextual. For example, the Chinese nationality is not vulnerable in terms of the world population, but can be a target of racism and xenophobia in non-Asian contexts.

Building on the notion of identity traits and target groups, let us look at some of the umbrella concepts regarding online hate speech within the Social Computing (CSS) community (Banko *et al.*, 2020; Balayn *et al.*, 2021):

- **Toxic speech.** It is an umbrella term (covering offence, hate, etc.) encompassing hurtful and aggressive content containing rude, disrespectful, or unreasonable language that is likely to cause people to leave the discussion (Dixon *et al.*, 2018).
- **Offensive speech.** Offensive speech often contains the use of derogatory and abusive phrases or hashtags, expressing inferiority towards the target entity.

[3] https://www.un.org/en/hate-speech/understanding-hate-speech/what-is-hate-speech

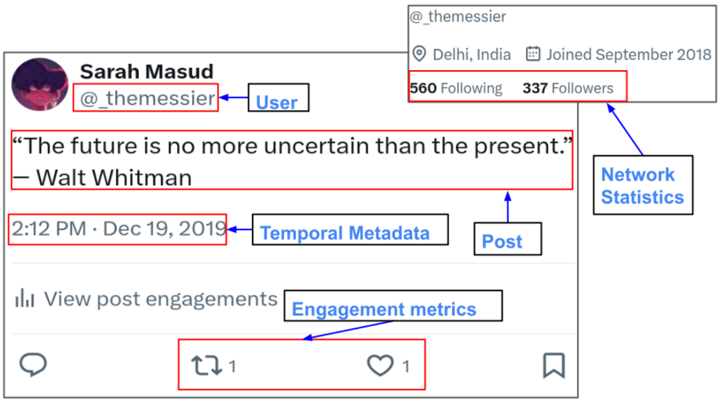


Figure 1.1: An example of online content as seen on X (taken from author's profile). The post forms the message that the user wishes to convey. Upon posting, other users on the platform can engage with the post, the volume of which is captured by the engagement metrics. The post also contains metadata information such as the date and time of posting. Also shown in the figure is an overview of the user's profile, with their network statistics highlighted.

- **Hate speech.** It is an extreme form of bias or negative stereotyping arising from distorted views about the target group that aims at silencing the target or promoting violence against them.
  It is essential to point out an important distinction between offence and hate. Anyone can offend anyone or anything. However, hate speech is recognised in terms of who the target is, as the notion of hate stems from a more deeply rooted prejudice. Thus, every hate speech is a specialised/subset form of offence or toxicity, but not vice versa (Balayn *et al.*, 2021). *Power dynamics is an essential characteristic of hate speech in the real world. However, it is hard to establish power dynamics in online spaces (Zhou* et al.*, 2023; Suler, 2004).*
- **Slur term.** A derogatory phrase whose negative connotation is widely understood by the public and is often employed to express toxicity. Like the concept of hatefulness, slur terms also possess dynamism and can be abusive or not, depending on when and who is using them against whom.
- **Explicit hate speech.** A form of hate speech that either contains slur term(s) or directly calls out the target community.
- **Implicit hate.** It is a coded form of hate speech that may not contain any explicit markers of hate but can still be contextualised as hateful (ElSherief *et al.*, 2021).
- **Implied stereotype.** The underlying/intended meaning coded in implicit hate.
- **Hate lexicon.** A dictionary of the slur and abusive terms employed for detecting explicit forms of hate speech. The mere presence of a slur term in a statement does not automatically make it hateful.
  It is interesting to note that not all abusive terms are slurs. Also, while slurs and hate lexicon contain semantically negative terms, on the surface and out of context, they can appear neutral. For example, the name of the pickle 'Kimchi' can be used to refer to Korean people in a derogatory manner.
- **Hate intensity.** It refers to an ordinal scale marking how explicit and harmful towards the target community a given hateful statement is (Masud *et al.*, 2022).

It is also crucial to familiarise oneself with the following OSN-related terminologies. An overview of the same is provided in Figure 1.1.

- **Online post.** A post is a piece of either human or machine-generated content that is available online for public consumption. The online post (referred to as a post henceforth) can be a series of unimodal texts and images or be multimodal (memes and videos). This thesis focuses on the textual form of content. While "post" is generally employed for any online content, from news articles to videos, some platforms also have a specific term for it. For example, a post on X is called a tweet, and on TikTok, it is called a reel. Consequently, on X, a retweet is when a tweet is reposted/reshared. In literature, it is usually prefixed with 'RT'.
- **Online user.** It refers to a digital profile (in our case, assumed to be a human) of someone engaging with the content on social networking platforms.
- **Root posts and users.** In literature as well as our thesis, we will refer to the *root post* or *source post* as the initial original post under consideration by our models. Consequently, the user posting them will be referred to as *root/source users*.
- **Online network.** Users can also connect with other users on a platform. Consider the case of $A \rightarrow B \rightarrow C$, where $\rightarrow$ means follows. Here, A will be privileged to access the content shared and endorsed by B. On platforms like X, these relations are directed, $A \rightarrow B \neq B \rightarrow A$. On Facebook, the connections are undirected, $A \rightarrow B = B \rightarrow A$. Moreover, A is a direct (1-hop) connection of B and a 2-hop connection of C, i.e., a friend-of-a-friend. In terms of OSNs, B is a 'follower' of C, and C is a 'followee' of B.
- **Interaction/Engagement metrics of a post.** These metrics are an indicator of the level of engagement garnered by a post, aka how many *'views'* it grabbed. It can be as simple as upvoting or downvoting via a *'like/dislike'* button on the interface or a more granular reaction from among '*like, support, cheer*', etc. The engagement can also be based on the number of comments and replies on the post.
- **Primary and auxiliary data.** While interactions, including comments, exist in relation to an online post, posts can exist as an independent entity. Hence, we refer to posts as the primary data. Meanwhile, the auxiliary or metadata consists of any information accompanying the post. For example, in our use case, auxiliary data can contain the time stamp of the post, the number of likes/replies, and the set of replies or comments, or even community flagging information like notes.

1.1.2 Content moderation

In a setup where users are allowed to post content on the platform, content moderation refers to a process of reviewing, flagging, and even removing content that is deemed harmful via the platform.

Extent of moderation. One can broadly categorise online platforms as moderated, semi-moderated, and unmoderated. Moderated platforms have a centrally defined content flagging policy that is applied uniformly for all intents and purposes, irrespective of the user's affiliation. Platforms like Facebook, Instagram, and X fall into this category. Semi-moderated platforms have a central but loosely defined moderation policy, the adoption of which is left to owners of the subspace or subgroup within the platform, where each subspace has a further granularly defined set of guidelines. Platforms like Reddit and Mastodon are examples of semi-moderated platforms. On the far end of the spectrum are unmoderated platforms like Gab, Parley, 4chan, etc, where users are free to share about anything with negligible repercussions.

Reactive and proactive moderation. As the name suggests, a reactive mechanism

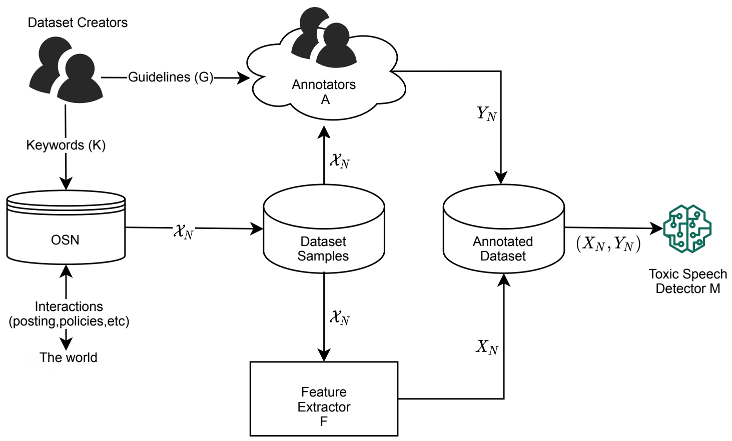


Figure 1.2: The pipeline for toxicity detection can be visualised as a sequence of data transformations (**Garg *et al.*, 2023**).: (i) the OSN sampler API ($S : K \rightarrow \mathcal{X}_N$) takes as input a set of keywords K and returns a set of samples $\mathcal{X}_N$; (ii) the annotation function ($A : \mathcal{X}_N, G \rightarrow Y_N$) converts $\mathcal{X}_N$ and the annotation guidelines G to a sequence of annotations Y_N; (iii) the feature extraction function ($F : \mathcal{X}_N \rightarrow X_N$) converts $\mathcal{X}_N$ to a sequence of feature vectors X_N. (iv) Finally, the function ($M : \mathcal{X}_N, Y_N \rightarrow Y'_N$) predicts the toxicity label.

for content moderation comes into play once content has been posted online and is publicly available. Proactive moderation, on the other hand, attempts to flag potentially harmful content even before it goes public. While reactive mechanisms are less intrusive, proactive mechanisms prevent the target community from being exposed to harm in the first place. Reactive strategies leave the decision-making in the hands of the content moderators; under proactive settings, the final content being posted is up to the user. Within the hate speech literature, both researchers and practitioners majorly work with reactive setups. Meanwhile, proactive techniques for offence detection are in the nascent stages of research and development (Katsaros *et al.*, 2022). Given that harm is not easy to encompass, the extent and type of content moderation on online platforms is solely based on the kind of engagement the platform wishes to engineer (Munn, 2020). In this thesis, we use content moderation to be broadly synonymous with toxicity moderation.

Role of computing in content moderation. It is well documented that content moderators are overburdened and mentally stressed by a deluge of content that needs filtering (Arsht and Etcovitch, 2018; Spence *et al.*, 2024; Steiger *et al.*, 2021). Computational techniques, especially natural language processing (NLP), are now being integrated with manual efforts (Spertus, 1997). From Application Programming Interfaces (APIs) and web crawlers to curating datasets to NLP techniques for data filtering and cleaning, and machine learning (ML) models for detecting hateful content, computation methods are employed throughout the pipeline. These techniques cover a range of tasks and analysis of hateful connotations from a statistical, linguistic, and topological point of view. Figure 1.2 outlines the toxicity detection pipeline (Garg *et al.*, 2023) adopted in literature and this thesis as well. From dictionary-based methods (Caselli *et al.*, 2020) to static vectoriser (Davidson *et al.*, 2017) to pretrained language model (PLM) based finetuning (Koufakou *et al.*, 2020; Caselli *et al.*, 2021*a*). Meanwhile, data annotation is largely human-driven (Rottger *et al.*, 2022). The role of large language models (LLMs) as both annotators and moderators under the n-shot setting is now being explored (Roy *et al.*, 2023).

1.2 Thesis motivation

The phrase "I know it when I see it" aptly applies to hate speech. While the categorical definition of hatefulness is subjective, the context around it provides necessary and often sufficient information to classify hate better. Especially within the pseudo-anonymous framework of online hate speech, we hypothesise that analysing and capturing the contextual signals at the right time can help us mitigate hate (Founta and Specia, 2021). It stems from the notion that via access to world knowledge and lived experiences, humans (directly or indirectly) have sufficient context to assess hatefulness. This knowledge is something that computational methods may lack. In such a case, the role of contextual signals becomes more prominent in the absence of explicit power dynamics in digital spaces.

1.2.1 Existing challenges in content moderation

As an initial step in developing our understanding as well as informing readers of the challenges and opportunities in the field of automated hate speech detection, we put together two surveys – one recognising the socio-technical challenges in modelling these systems (**Chakraborty and Masud, 2022**) and the other exploring the downstream biases once the systems are deployed (**Garg *et al.*, 2023**). We expand upon these issues and employ them as a motivator for the thesis.

Dynamic contextualisation. Sentiment and emotion can be termed as the underlying sources or driving forces towards hatred. The development of hateful sentiments affects the production of hate speech, which in turn affects hate crimes. While hate speech is semantically similar to expressing negative emotions (Zhou *et al.*, 2021*b*), reducing the task of classifying hate speech to sentiment or emotion detection takes away from the fact that hate is not just a notion (like positive or sad) but rather a bias-motivated behaviour (Pretus *et al.*, 2023; Vendrell Ferran, 2024). There are two significant drawbacks of directly using sentiment as a feature. Firstly, marking emotions becomes difficult when hate speech is implicit. Here, the sentiment can be positive or neutral, while hate speech is still present. Secondly, unlike sentiment analysis, which has a representation in GLUE benchmarking (Wang *et al.*, 2019), the equivalent does not exist for hate detection. It is imperative to point out that this is not by design but because of the dynamic nature of hate speech that renders static benchmarking redundant.

The ephemeral nature of hate speech on social media (explicitly viral content often gets taken down), coupled with the lack of standard benchmarks, forces the researchers to introduce newer datasets for a specific context, language, or modality, often with distinct yet semantically overlapping class labels (Fortuna *et al.*, 2020). Further, the recent removal of free API access to OSNs has led to a reduction in curating newer datasets or updating older ones. We face these challenges in our thesis work as well. Another persistent challenge in hate speech literature is the annotation bias. Human evaluators from diverse backgrounds are necessary to provide coverage against hate speech. However, the background of the evaluators also contributes to annotation biases (Rottger *et al.*, 2022; Aroyo *et al.*, 2019*a*; Munn, 2020). Analysis and mitigation of biases in hate speech is an active area of research (Garg *et al.*, 2023; Biester *et al.*, 2022; Wojatzki, 2018; Sap *et al.*, 2019). Succinctly, with no sacrosanct demarcation of hatefulness, it is tricky to codify into a fixed paradigm. This variation introduces social, legal, and technical challenges when computational solutions are deployed for hate detection (Van Alstyne, 2024; Van Alstyne *et al.*, 2023; Parker and Ruths, 2023).

In terms of NLP-driven moderation, based on our initial survey (**Chakraborty and Masud, 2022**), the thesis identifies four significant gaps, focusing on textual modality.

- First of all, unlike real-world interactions, determining both intent and power dynamics in online spaces is intractable (Davani *et al.*, 2025; Hovy and Yang, 2021). Coupled with limited API access, this forces the majority of hate detection systems to employ solely the content of the post in isolation to assess its hatefulness. ***It begs the question of context-centric data curation and modelling w.r.t both hateful posts and users.***
- Secondly, the practice of assessing automated content moderation under a classification setup does not fully utilise the generative nature of language models (LMs). Under both proactive and reactive settings, a natural and interactive setup can help both users and content moderators make a more informed decision. ***This reinforces the development of human-centric content moderation tools and techniques.***
- Thirdly, the current rate at which hateful content is published is difficult for humans to tackle alone. While human moderation is of utmost importance in combating hate speech, it comes at the cost of psychological burdens that often go unnoticed and uncompensated (Steiger *et al.*, 2021). ***There exists a gap in the alignment between real-world personas and automated content moderation tools.***
- Lastly, following the vogue of NLP, the majority of efforts in hate speech literature have focused on (a) direct/explicit forms of hate and (b) English-based text and Western perceptions of hatefulness. Among the plethora of hate speech datasets, only a handful are focused on either Hinglish content (code-mixed Hindi and English) or implicit hate labels (Vidgen and Derczynski, 2021). ***It calls for investment towards multilingual and implicit benchmarking.***

While mentioning the above computational challenges related to toxicity and hate speech detection, one must acknowledge that in the deployment and engagement of NLP/ML models in the real world, the legal implications around hate speech and content moderation are as ill-formed as the definition of hate speech (Van Alstyne, 2024). Given the lack of expertise, a discussion of legality is beyond the scope of this thesis; it does highlight that examining hatefulness is a very human-centric and multi-stakeholder phenomenon with social, economic, and legal implications (Jiang *et al.*, 2023*b*).

1.2.2 What are contextual signals?

Throughout the thesis, we define **contextual signal** as any form of additional/auxiliary information that can be provided along with the input text of the post to offer more *'context'*, nudging the base model to be more *'toxicity attuned.'* Going back to the example in Figure 1.1, metadata statistics such as the user's network statistics or the post's engagement metrics are one example of the contextual signal. It stems from the observation that differences in the level and frequency of engagement can be one indicator of hatefulness (Founta *et al.*, 2019; Mathew *et al.*, 2019). Another set of contextual signals is the topological information about the set of users who interact with a post. Like begets like, and hateful users are more likely to follow other hateful users

(Goel *et al.*, 2023). The platform-specific auxiliary signals discussed so far will hereafter be referred to as *endogenous signals*. However, these signals can also be obtained outside the platform. Any information that is not directly obtained from the platform is termed an *exogenous signal*. Crawled news articles (Dutta *et al.*, 2020), knowledge extracted entities (Lin, 2022), or even other LLM prompted outputs (Sridhar and Yang, 2022) are some examples of exogenous signals.

The endogenous and exogenous signals we have enlisted so far do not require any further annotations. However, contextual signals can also be sourced from secondary annotations. As the standard practice is to annotate the post's content for hate or not, secondary annotations can include, for example, the intensity of hate, the type of explicitness, and the target group of the hate (ElSherief *et al.*, 2021). As these are annotated along with the samples, they are termed as *in-dataset attributes*. In case another external model annotates them, but within the context of hate speech, they will be referred to as *in-domain attributes*. Both in-domain and in-dataset contextual signals are toxicity attributes (AlKhamissi *et al.*, 2022) in that they augment the information in an input text with toxicity-related signals. Note, both *in-dataset* or *in-domain* "toxicity attributes" can be either endogenous or exogenous, the main criteria being not their platform affinity but task specificity and hatefulness, in our case, for example.

Apart from the taxonomy of (endo/exo)genous and in-(dataset/domain), contextual signals can also be classified based on whether they are 1-1 mapped to the input text or indirectly connected. For example, a post's metadata is intrinsically tied to the post and varies from post to post by the same user. Even endogenous signals, like trending hashtags or news of the day, need not be explicitly associated with a post. However, they can still be a latent influence on the post's genius or engagement (Myers *et al.*, 2012).

In this thesis, we will explore all the above contextual signals, sometimes multiple signals in tandem, to assess hatefulness better. Note that the taxonomy discussed above is not absolute and complete, and will likely grow in the future as modes of communication and human interaction change. Also, the taxonomy described above is not isolated and exclusive; for example, a signal can be endogenous and 1-1 mapped or any combination thereof.

1.3 Thesis organisation and contributions

Consolidating the research gaps outlined in Section 1.2.1, the motivation for the thesis can be summarised as follows:

> The thesis quantifies the role of contextual signals to aid the content moderation pipeline intersecting multilingual and implicit aspects of hatefulness.

Keeping the classical NLP paradigm as our cornerstone for both classification and generation tasks, *the thesis uncovers what contextual signals are best suited for which hate speech-related tasks and where in the modelling pipeline they should be introduced.* In this thesis, we limit our scope to textual content.

The thesis is divided into three parts, with Chapter 2 serving as a common literature review. A list of abbreviations employed in the thesis is provided in Tables 1.3 and 1.4.

- **Part I: Establishing the importance of contextual signals**
  - **Chapter 3.** This chapter introduces contextual signals via an English-text dataset consisting of primary and auxiliary signals curated from trending Indian politics on X (Twitter). The ConInHate (Contextual Indian Hate) dataset, consisting of 30k tweets cascades and 13k user profiles spanning 34 hashtags, is collected and manually annotated (23k tweets) for binary hate/non-hate labels. This labelling is later employed to develop RETINA (Retweeter Identifier Network with Exogenous Attention), which helps predict the retweet behaviour of hateful users. Our work (**Masud *et al.*, 2021**) highlights the need for the inclusion of exogenous and endogenous signals towards building retweet predictors. Even under skewed class ratios, our signal-rich system remains robust, achieving a macro-F1 of 0.85 against best information diffusion baselines.
    Working with ConInHate and featured-engineered RETINA exposes two significant research gaps: (i) the need for a systematic review of PLM-based hate detection systems and (ii) the need for culture and language-specific hate speech datasets. After reviewing the plethora of English language datasets, an extensive analysis of the effectiveness of the PLM finetuning paradigm was performed through five PLMs, seven hate speech datasets, and four research settings. In a one-of-a-kind study (**Masud *et al.*, 2024*b***), we discover how finetuning dynamics for hate detection differ from standard NLP tasks, calling into question the role of domain-specific subjective PLMs. The findings are employed in a novel neutrally-seeded dataset with a focus on multilingualism. The GOTHate (Geo-pOlitical Topical Hate) dataset additionally consists of endogenous signals in the form of the user's timeline and network information. The manually annotated $50k$ codemixed Hinglish tweets are then augmented with unsupervised endogenous user and post's signals. GOTHate is employed to develop HEN-mBERT (History-Examplar-Network Infused mBERT) for four-way hate speech detection. Incorporating the findings from both contextual signalling and the PLM's finetuning examined earlier in this chapter, HEN-mBERT achieves a macro-F1 of 5% in hate class over vanilla mBERT upon finetuning (**Kulkarni *et al.*, 2023**).
    As a byproduct of working with network data modelling for both RETINA and HEN-mBERT in this chapter, we notice a gap in the availability of large-scale network visualisation tools. The chapter ends with a discussion of DiVA (Diffusion Visualisation and Analysis) – a tool for visualising information diffusions (**Sahnan *et al.*, 2022**). Developed over multiple iterations, the web tool allows side-by-side comparison of two diffusion modelling setups, even supporting analysis of custom diffusion algorithms in Python.
  - **Chapter 4.** Despite a fine-grained annotation of hate speech with GOTHate and HEN-mBERT, there still exists a gap in modelling the notion of implicitness, which is the focus of this chapter. However, curating implicit hate speech from social media via a keyword or topic-based pipeline employed in the last chapter is not feasible, given the subtle language of implicit hate. As a stepping stone to better understand the nature of implicitness, we look into political attacks, which are relatively easier to monitor. Using the setup of political attacks as a subcategory of offensive and toxic content, we track tweets by parties and ministers in India during the February 2020 assembly elections. To this end, we curate a dataset of 46k Hinglish political attacks

with 1.7k samples manually annotated as implicit, explicit, or neutral (non-attacking). `PolAt` (Political Attack) helps quantify the power dynamics of major national parties, as well as the variability in self-promotion and negation compared to the incumbent party in the states. The preliminary analysis of implicitness via `PolAt` reveals a lack of contextual modelling of implicit hate via PLMs (**Masud and Chakraborty**, **2023**).

Expanding our manual analysis of implicitness to its understanding in the latent space, we empirically establish the closeness of implicitly hateful samples to non-hateful ones. The observations point towards the need for auxiliary signals that operationalise this push-pull between the semantic similarity of implicit hate with neutral content, but intended similarity with hatefulness. We hypothesise the use of "free text explaining the intended hate" as the auxiliary signal that fits the need. To this end, `FiADD` (Focused inferential Adaptive Density Discrimination) is proposed to augment the three-way classification of hate (explicit, implicit, or neutral) (**Masud *et al.***, ***2024a***). It employs implied explanations as a contextual signal to bring the surface and hidden form of implicitness closer in the latent space, improving its detection. Due to the ease of availability of such explanations in English, `FiADD` is tested on three English (western-centric) implicitness datasets.

Concluding the first part of our thesis, we note the unintended biases in toxicity detection literature. We observe in our survey (**Garg *et al.***, **2023**) that the implications of the biases go beyond the standard metrics in hate speech detection, whether binary or three-way.

- **Part II: Developing human-centric content moderation tools**
  - **Chapter 5.** The classification-based contextual hate speech detection systems (Part I) evaluated for the first level of filtering have three major shortcomings. Firstly, it does not help to explain in a human-friendly manner why a data point is considered hateful. Secondly, it fails to utilise the generative nature of LMs. Lastly, the endogenous and exogenous signals discussed in the last two chapters, while pertinent to hate speech research, are difficult to curate at scale. Instead, this chapter explores the development of human-centric content moderation tools by employing in-dataset and in-domain contextual signals. The chapter begins by reevaluating a popular technique for external knowledge infusion in literature, the knowledge graph (KG) tuples. Based on our experience of handling implicit content in Chapter 4, we postulate that the process of obtaining KG tuples based on surface-level semantic similarity may not lend itself to task specificity to improve explanations of implicitness. (**Yadav *et al.***, **2024**). Working with two datasets and two KGs, we record that the replacement of top-k tuples by a random-k set as auxiliary signals does not lead to the expected performance deterioration.

    Motivated by the confounding observation, this chapter delves into the notion of "toxicity attributes." Given implicit hate as input and its toxicity attributes as input signals, `Tox-BART` (Toxicity-attributed BART) generates explanations in natural language to elicit the underlying implied stereotype. Across automated and human evaluations, we observe that finetuned `Tox-BART` provides more understandable and target-specific explanations, even surpassing zero-shot prompting. The specific, albeit explicit, explanations when visible only to content moderators (human or computational) can help assess the harmfulness of an implicit post (**Yadav *et al.***, **2024**).

- **Chapter 6.** The set of both classification tasks (Chapters 3, 4) and the generative task (Chapter 5) explored so far operates under a reactive setup. Reactive mitigation of hate puts both power and responsibility in the hands of content moderators, and this top-down approach puts social media users at the bottom of the pipeline, which can act as a deterrent to free expression. Navigating the grey area between freedom of speech and safe spaces, this chapter explores the proactive and user-nudged mitigation of hate. Our aim is not to render a hateful message into a non-hateful one, as a shift this big can backfire. Instead, we hypothesise that a reduction in hate intensity, while maintaining its explicitness, can be one way of achieving a middle ground (**Masud *et al.*, 2022**). To assess the impact of such a modification, we manually curate a novel corpus of explicitly hateful samples in English and their less intense counterparts, a.k.a the `HateNorm` (Hate Normalisation) dataset. Employing `HateNorm`, we finetune BART under a discriminative reward aimed at reducing the predicted intensity of the generated text. We perform an extensive automated, human, as well as in the wild analysis to establish the deployability of `NACL` (Neural hAte speeCh normaliser) in the real world proactive mitigation.

- **Part III: Future direction of human-LLM alignment for mitigating hate**

  - **Chapter 7.** Examination of various tasks and datasets for hate speech modelling in the thesis reinstates the importance of both human annotators to produce contextually specific annotations as well as the role of context. Yet both context signals and culturally attuned human annotations are hard to reproduce. Working closely with ephemeral social media datasets, the human evaluators, and crowdsourced workers in previous chapters, we can safely attest to the ordeal of research in hate speech. Notwithstanding the fact that human annotations for hate speech are rife with annotation biases that lead to downstream harm (**Garg *et al.*, 2023**). Despite the issues in human annotations, their lived experiences are pertinent to hate labelling. On the other hand, the rise of the now-famous zero-shot prompting via LLMs calls into question the role of LLMs in supporting crowdsourced annotations for hate speech. We thus conclude the thesis with an assessment of how LLMs can better augment the human annotation process, rather than pushing for the replacement of humans by LLMs. Our analysis (**Masud *et al.*, 2024*c***) starts with an annotation setup consisting of one human and one LLM annotator (working independently of each other). We then introduce context cues/signals in zero-shot prompting to map the influence of geographical, demographical, and numerical anchors in LLM. We examine these three factors as they closely mimic culture-specific human annotation. Reflecting on the fundamental changes in our lives with the introduction of Generative AI (GenAI) interfaces, we conclude our thesis with a broader discussion of what lies ahead in terms of judging the output from GenAI for subjective prompts/inputs. We also comment on the role of users of GenAI in evaluation of the output's utility, looking beyond standard ML metrics (**Chakraborty and Masud, 2023, 2024**).

1.4 Research impact

Overall, the thesis aims to help content moderators, social media users, and researchers in the following ways:

- **Content moderators.** The thesis provides a two-fold support to moderators:
  1. **Reducing burden.** The majority of explicit leaning hate speech can be effectively detected by up-to-date and contextual hate speech detection models. By aiding first-level automated flagging, one can reduce the burden of human moderators witnessing explicitly harmful content.
  2. **Enabling explanations.** As implicit hate is hard to detect by both humans and models, we discover ways of providing explanations of underlying stereotypes/implied meanings for better detection of implicit hate.
- **Social media users.** Employing a nudge-based model leaves the final decision in the hands of users while also making them aware of alternative and less intense semantics for expressing the intended negativity. Nudging online users to proactively reduce hatefulness not only leads to detoxification of the overall system (Dementieva *et al.*, 2025) but also brings agency to users, allowing for corrective behaviour via a bottom-up approach.

Table 1.1: A summary of datasets curated as a part of this thesis, highlighting whether the dataset contributes to multilingualism, implicitness, or contextual signals. It also highlights whether the dataset primarily focuses on Indian socio-political topics or not. C/G stands for classification/generation, enlisting the type of NLP task the dataset is suited for. * refers to existing datasets/hate samples for which the thesis provides additional annotations. For ++ datasets, the # human annotations are not the same as the overall dataset size.

Datasets (↓)/Feature(→)	**Multilingual**	**Indian**	**Implicit**	**#Human Annotations**	**Auxillary Signals**	**Task**
`ConInHate` (**Masud *et al.*, 2021**)	✔	✔	✖	≈23k++	Engodenous, Exogenous	C
`GOTHate` (**Kulkarni *et al.*, 2023**)	✔	✔	✖	≈ 50k	Engodenous	C
`PolAt` (**Masud and Chakraborty, 2023**)	✔	✔	✔	≈1.7k++	✖	C
`HateNorm*` (**Masud *et al.*, 2022**)	✖	✖	✖	≈3k	In-dataset	C+G
`ImpGab*` (**Masud *et al.*, 2024*a***)	✖	✖	✔	≈400	In-dataset	G
`AbuseEval*` (**Masud *et al.*, 2024*a***)	✖	✖	✔	≈800	In-dataset	G

Table 1.2: A summary of models/frameworks developed as a part of this thesis, highlighting whether the system tackles implicitness and proactiveness (or reactive). C/G/V stands for classification/generation/visualisation as the type of task performed by the respective system.

Model (↓)/Feature(→)	**Implictness**	**Proactive**	**Task**
`RETINA` (**Masud *et al.*, 2021**)	✖	✖	C
`HEN-mBERT` (**Kulkarni *et al.*, 2023**)	✖	✖	C
`FiADD` (**Masud *et al.*, 2024*a***)	✔	✖	C
`Tox-BART` (**Yadav *et al.*, 2024**)	✔	✖	C+G
`NACL` (**Masud *et al.*, 2022**)	✖	✔	C+G
`DiVA` (**Sahnan *et al.*, 2022**)	-	-	V

- **CSS researchers/practitioners.** The thesis provides useful resources and insights for practitioners to build on. Links to code and datasets produced as a part of the thesis are listed in Appendix J. The contributions are encapsulated via:

1. **Datasets.** A primary contribution of this research is providing large-scale multilingual toxicity detection datasets covering the Indian Twitter/X[4]. Each dataset is augmented with contextual signals along with human annotations. Apart from datasets covering the Indian context, we also enhance the annotations (or lack thereof) of 3 existing hate speech datasets (2 of which are implicit). Table 1.1 summarises the dataset contributions.
2. **Modelling.** For datasets varying in size, scope, and context, the thesis empirically quantifies which contextual signals are best suited for which content moderation task. Not only does the research highlight the importance of including contextual signals, but it also remarks on when and how they should be infused into the modelling to mimic human decision-making better. Negative results suggest that no single augmentation is all-encompassing. Table 1.2 summarises the modelling contributions.
3. **Deployment practises.** Through exhaustive and repeated observations, the research enlists the best practices when finetuning or prompting language models for hate speech detection. The thesis follows the development of language modelling in NLP, from doc2vec (Le and Mikolov, 2014) to finetuning BERT-based PLMs (Vaswani *et al.*, 2017; Devlin *et al.*, 2019). For certain use cases and baselines, zero-shot LLMs (Kojima *et al.*, 2022) are also evaluated. The research reinforces that LLMs are not one-size-fits-all, as not all tasks require LLMs, and neither are LLMs a panacea for entirely replacing humans in content moderation.

Table 1.3: Abbreviations of systems used in the thesis.

Type	Abbreviation	Full form
Platform	OSN	Online Social Network
Paradigm	API	Application Programming Interface
Paradigm	AI	Artificial Intelligence
Paradigm	GenAI	Generative AI
Paradigm	NLP	Natural Language Processing
Paradigm	LM	Language Model
Architecture	FNN	Feedforward Neural Network
Architecture	CNN	Convolutional Neural Networks
Architecture	GRU	Gated Recurrent Unit
Architecture	LSTM	Long Short-Term Memory
Architecture	Bi-LSTM	Bidirectional Long Short-Term Memory
Architecture	RNN	Recurrent Neural Network
Paradigm	PLM	Pretrained Language Model
Paradigm	LLM	Large Language Model
Architecture	BERT	Bidirectional Encoder Representations from Transformers
Architecture	mBERT	Multilingual BERT
Architecture	BART	Bidirectional and Auto-Regressive Transformers
Dataset	ConInHate	Contextual Indian Hate
Architecture	RETINA	Retweeter Identifier Network with Exogenous Attention
Dataset	GOTHate	Geo-pOlitical Topical Hate
Architecture	HEN-mBERT	History-Examplar-Network Infused mBERT
Tool	DiVA	Diffusion Visualisation and Analysis
Dataset	PolAt	Political Attacks
Architecture	FiADD	Focused Inferential Adaptive Density Discrimination
Architecture	Tox-BART	Toxicity-attributed BART
Dataset	HateNorm	Hate Normalisation
Architecture	NACL	Neural Hate Speech Normaliser

[4]Twitter is now X, but at the time of datasets being curated, it was Twitter. We interchangeably reference them in this thesis.

Table 1.4: Abbreviations of evaluation metrics used in the thesis.

Type	Abbreviation	Full form
Classification	P	Precision
Classification	R	Recall
Classification	macro-F1	Macro-F1 Score
Classification	ACC	Accuracy
Classification	AUC	Total Area Under the Curve
Classification	ROC	Receiver Operating Characteristic
Classification	IAA	Inter-annotator Agreement
Classification	PHLR	Predicted Hate Label Ratio
Generation	BLEU	Bilingual Evaluation Understudy
Generation	ROUGE	Recall-Oriented Understudy for Gisting Evaluation
Generation	BERTScore	BERTScore
Generation	Perplexity	Perplexity
Regression	Pearson	Pearson Correlation Coefficient
Regression	Spearman	Spearman's Rank Correlation Coefficient
Regression	MCC	Matthews Correlation Coefficient
Regression	RMSE	Root Mean Squared Error
Ranking	MAP@K	Mean Average Precision at K
Ranking	HITS@K	Hit Rate at K
Similarity	CS	Cosine Similarity
Similarity	JS	Jensen–Shannon Divergence
Similarity	MMD	Maximum Mean Discrepancy
Similarity	ALD	Average Linkage Distance
Similarity	ACLD	Average Centroid Linkage Distance

CHAPTER 2

Related Work

"Scientific knowledge is a body of statements of varying degrees of certainty – some most unsure, some nearly sure, none certain."
- Richard P. Feynman; The Value of Science

Over the years, instances of distress (Saha *et al.*, 2019), cyber-bullying (Agrawal and Awekar, 2018), and even offline violence (Williams *et al.*, 2019) have arisen out of online harmful behaviour. This chapter enlists the related work referenced throughout this thesis.

2.1 Online toxicity detection

A simple endorsement by an individual or expression of superiority is not hatred, even though the endorsing individual is more prone to participating in hate speech. Studies have shown that a combination of the platform's system design (Munn, 2020), content moderation policy (Lima *et al.*, 2020), and quality of discourse (Wang *et al.*, 2021) can support/curb the spread of hateful content like a self-fulfilling prophecy. When hateful users get banned from platforms like X, Facebook, YouTube, and Reddit, they often find a home in less restrictive, alt-right platforms such as Gab (Than *et al.*, 2020; Johnson *et al.*, 2019). Such users frequently also collude in close circuits to quickly propagate their ideas or act more viciously (Massanari, 2017; Chatzakou *et al.*, 2017).

2.1.1 Datasets

Explicit hate. Unlike other subjective tasks like sentiment analysis, toxicity does not have a GLUE equivalent (Wang *et al.*, 2019) benchmark (Piot *et al.*, 2024; Tonneau *et al.*, 2024). Even though some linguistic test cases have been proposed for hate speech, they do not encompass the dynamic and semantic diversity of hatefulness (Röttger *et al.*, 2021; Kirk *et al.*, 2022; Röttger *et al.*, 2022). The lack of standardisation has led to an avalanche of toxicity-related datasets (Laaksonen *et al.*, 2020; Founta *et al.*, 2018). Based on the hypothesis that gender and race are primary targets of hate speech, Waseem and Hovy (2016) released a dataset of $16k$ tweets labelled in-house. Davidson *et al.* (2017) released a crowd-sourced annotated dataset of $25k$ tweets. Along similar lines, we have hate and offence datasets from diverse online forums like Wikipedia (Wulczyn *et al.*, 2017), Stormfront (de Gibert *et al.*, 2018), Facebook (Salminen *et al.*, 2018), Reddit (Mollas *et al.*, 2022), etc. Later Founta *et al.* (2018) provided a large-scale corpus of $80k$ English tweets curated via Twitter streaming API. However, they applied bootstrapped sampling to enhance the volume of minority classes. However, most curated datasets favour explicit forms of hate (Wiegand *et al.*, 2021). Meanwhile, neutral seeding does not rely on a specific slur term. Instead, it looks at a broader socio-political topic that is controversial and will potentially contain offensive statements around the topic. While a few neutrally seeded datasets also exist (de Gibert *et al.*, 2018; Basile *et al.*, 2019), they lack coverage of events and target groups. For specific target groups and types of hate speech, such as sexism (Kirk *et al.*, 2023) or xenophobia (Sánchez-Junquera *et al.*, 2021), researchers have also explored multi-level annotations, allowing for more structured and linguistic analysis (Merlo *et al.*, 2023) of toxicity.

Implicit hate. Studies have been conducted to understand the subtle forms of implicit hate. ConvAbuse consists of 4k English samples obtained from in-the-wild human-AI conversations with AI chatbots. Each conversation is marked for the degree of abuse (1 to 3) and directness (explicit or implicit). On the other hand, `ImpGab` (Kennedy *et al.*, 2022) consists of 27k posts from Gab, which contain a hierarchy of annotations about the type and target of hate. Similarly, `AbuseEval` (Caselli *et al.*, 2020) consists of 14k Twitter posts augmented with 'explicit' and 'implicit' labels. `LatentHatred` (ElSherief *et al.*, 2021) is, however, the most extensive and most widely used implicit hate speech dataset. It consists of $21k$ Twitter samples labelled for implicit hate as well as 6 additional categories of implicitness. It also contains free-text human annotations explaining the implied meaning behind the implicit posts. Along similar lines, SBIC (Ocampo *et al.*, 2023*c*) is also a collection of 44k implicit posts curated from online platforms with human-annotated explanations. However, SBIC does not have a direct marker for the explicitness of the post, and by default, all posts are implicit hate. More recently, the ISHate (Ocampo *et al.*, 2023*c*) dataset has been curated by combining existing hate speech and counter-hate speech datasets and relabelling the samples for explicit-implicit markers, consisting of 30k samples labelled as explicit, implicit, subtle, or non-hate. It is interesting to note that in their analysis, the authors do not showcase how the different datasets interact with each other in the latent space. We hypothesise that the performance improvements are obtained not as a result of modelling but due to the fact that these samples are obtained from different datasets. Meanwhile, ToxiGen (Hartvigsen *et al.*, 2022) is a curation of around 1M implicit hate statements generated by prompting LLMs.

Multilingual and multimodal. To improve coverage, a broader set of languages (Vidgen and Derczynski, 2021) and hate speech-related tasks are now being considered. Depending on the language of the current word, the meaning conveyed by a code-mixed sentence can change, which poses additional challenges for collecting and annotating datasets under code-mixed settings. On the other hand, it is hard to develop statistically sound systems for low-resource and code-mixed content without training on large-scale data (Ranasinghe and Zampieri, 2021). Researchers are now focusing on datasets specific to diverse languages (Mulki *et al.*, 2019; Pavlopoulos *et al.*, 2017; Romim *et al.*, 2021; Mandl *et al.*, 2019; Rizwan *et al.*, 2020). Supervised datasets in medium and low-resource languages like Arabic (Alshaalan and Al-Khalifa, 2020), German (Assenmacher *et al.*, 2021), Hindi (Chopra *et al.*, 2020; Bohra *et al.*, 2018), Indonesian (Alfina *et al.*, 2017), Urdu (Rizwan *et al.*, 2020), Bengali (Karim *et al.*, 2021), Vietnamese (Vo *et al.*, 2025) etc., to name a few, are publicly available. DALC (Caselli *et al.*, 2021*b*) is a Dutch dataset consisting of 8k tweets curated from Twitter, labelled for the level of explicitness as well as the target of hate. Recently, Toraman *et al.* (2022) proposed another large-scale crowd-sourced dataset for English and Turkish (100k each), curated across five topics with manually curated keywords to sample the posts.

Online communication exists in varying forms apart from texts, such as emojis (Saikh *et al.*, 2024), images, audio-video (Wang *et al.*, 2024), or a combination of them. These different modalities also provide varying cues about the message. Memes are specifically emerging as a source of expressing hate while circumventing explicit hate detection systems (Kiela *et al.*, 2020; Oriol *et al.*, 2019; Sharma *et al.*, 2022; Pramanick *et al.*, 2021). Building upon the multimodal hate meme dataset MMHS150K (Gomez *et al.*, 2020), a multimodal implied hate dataset (Botelho *et al.*, 2021) has been proposed.

2.1.2 Detection via classification

Given the overwhelming work of manually flagging hateful content, the research community has been looking at methods to aid human moderators with the automatic flagging of hateful content. Work by Schmidt and Wiegand (2017); Vidgen and Derczynski (2021); Tahmasbi and Rastegari (2018) provide independent systematic surveys of the datasets and models in hate speech detection. In the majority of the use cases, the task of hate speech detection is treated as a classification problem (either binary or multiclass). Initially, simple n-gram feature-based (lexical and textual) Logistic Regression models of hate speech detection were proposed. Starting from there, ensemble feature engineering methods gained prominence (de Gibert *et al.*, 2018; Gao and Huang, 2017). A significant shift in the study of hate speech detection occurred with the use of Glove-based embedding (Pennington *et al.*, 2014) instead of the n-gram models. Non-contextual embedding was combined with vanilla CNN, LSTM (Badjatiya *et al.*, 2017). In a similar vein, a variety of automatic online hate speech detection models have been proposed across languages (Fortuna *et al.*, 2019; Founta *et al.*, 2019; Rizwan *et al.*, 2020).

However, with the arrival of the transformer architecture (Vaswani *et al.*, 2017), hate speech tasks also gained a significant boost (Mathew *et al.*, 2021; Caselli *et al.*, 2021*a*; Masud *et al.*, 2022; Mozafari *et al.*, 2020*a*; Ghosh and Senapati, 2022). Consequently, PLMs like HateBERT (Caselli *et al.*, 2021*a*) and BERTweet (Nguyen *et al.*, 2020) have also been proposed, built upon the concepts of continued pretraining on top of BERT (Devlin *et al.*, 2019). In the case of HateBERT, the corpus for performing masked language modelling (MLM) was obtained from subreddits that are potentially offensive and are likely to contain hateful terms. On the other hand, in the case of BERTweet, a large corpus of tweets was obtained from the general Twitter stream. In both HateBERT and BERTweet, the corpus is unlabelled and allows for domain adaptation towards toxic and informal social media language patterns, respectively.

Nowadays, LLMs can perform complex tasks with the help of demonstrations (Liu *et al.*, 2022*b*) via in-context learning (ICL) (Brown *et al.*, 2020) and prompt engineering (Singhal *et al.*, 2022). Catching up with the recent trends in this space, the role of demonstrations and prompting has been examined in hate speech as well (Huang *et al.*, 2023*a*; Yang *et al.*, 2023*b*). From zero-shot (Plaza-del Arco *et al.*, 2023; Nozza, 2021), few-shot (Chiu *et al.*, 2022), chain-of-thought (AlKhamissi *et al.*, 2022) prompting to proactive nudging (Agarwal *et al.*, 2023). However, in-context prompt examples are prone to sampling bias (Zhao *et al.*, 2021; Zamfirescu-Pereira *et al.*, 2023) and impacted by the quality of exemplars (Min *et al.*, 2022; Liu *et al.*, 2022*a*; An *et al.*, 2022; Lyu *et al.*, 2023).

In the case of implicit hate detection, studies have explored infusing context in the form of knowledge graph (KG) tuples (ElSherief *et al.*, 2021) or Wikipedia summaries (Lin, 2022). Other lines of work have explored data augmentation techniques (Ocampo *et al.*, 2023*c*). From the latent space perspective, researchers have studied how the infusion of a common target group can bring explicit and implicit samples closer (Ocampo *et al.*, 2023*b*). While the idea is intuitive since implicit hate and explicit slurs are specific to a target group, here, the extent of overlap in the case of multiple target identities is not well defined.

Loss function. Cross-entropy loss (CE) is the most widely used loss function for classification tasks, including hate speech. However, it impacts the inter/intra-class clusters sub-optimally (Liu *et al.*, 2016). As classification tasks can be modelled as obtaining distant clusters per class label, one can exploit clustering and distance-metric approaches to enhance the boundary among the class labels, leading to improved classification performance. The most popular deep metric learning is the contrastive loss family (Chopra *et al.*, 2005; Schroff *et al.*, 2015; Chen *et al.*, 2017). However, like CE, contrastive loss operates on a per-sample basis. In order to benefit from the one-to-one mapping of the implicit hate and its implied meaning, contrastive learning has been explored (Kim *et al.*, 2022), which has only provided slight improvement. At any given point, hateful samples (either explicit or implicit) form less than 10% of the content present on the web. To overcome the class imbalance, data augmentation (Venturott and Ciarelli, 2020; Roychowdhury and Gupta, 2023) and adversarial techniques (Cao and Lee, 2020) have been tested.

Generalisability and debiasing. Deep learning models are criticised for being black boxes. While heuristics such as LIME (Ribeiro *et al.*, 2016) and SHAP (Lundberg and Lee, 2017), among others, attempt to make the classification output interpretable, they are limited to perturbations in the input space rather than the latent space. More recently, work on mechanistic interpretability (Elhage *et al.*, 2021) attempts to understand how transformers build their predictions across layers. Interpretability (Vijayaraghavan *et al.*, 2021) and sufficiency (Balkir *et al.*, 2022) in hate speech have always been open research areas. Moreover, issues with generalisation (Yin and Zubiaga, 2021) and biasing (Garg *et al.*, 2023; Balayn *et al.*, 2021; Biester *et al.*, 2022; Wojatzki, 2018; Sap *et al.*, 2019) in both PLMs and LLMs are prevalent and active research areas. Human annotations for subjective tasks (Rottger *et al.*, 2022) are rife with ambiguity (Kanclerz *et al.*, 2022; Aroyo *et al.*, 2019*b*) and biases (Garg *et al.*, 2023). Of particular interest is the annotation bias (Wich *et al.*, 2021). In hate speech, annotation bias manifests due to differences among the annotators' beliefs (Sap *et al.*, 2022), experience, world knowledge (Yin and Zubiaga, 2021; Abercrombie *et al.*, 2023), and social-demographic conditioning (Orlikowski *et al.*, 2023). Disparity in access to additional context (Ljubešić *et al.*, 2022; İhtiyar *et al.*, 2023) and annotation guidelines (Ross *et al.*, 2016), in some instances, reduces the bias and, in some cases, confirms the annotator's biases. Researchers have not only empirically proved how the classifiers are biased to classify tweets by African American Dialect (Sap *et al.*, 2019; Davidson *et al.*, 2019) as hateful but also how the intersection of multiple unintended biases (Huang *et al.*, 2020) like gender, age, and geography can also lead to false positives in the detection (Zhou *et al.*, 2021*a*). Debiasing of hate speech classifiers is crucial. While some suggest replacing identity marker/stereotypical terms with their generic WordNet (Miller, 1995) parent (Badjatiya *et al.*, 2019), others perform external regularisation (Kennedy *et al.*, 2020). Parallel research seeks to model diverse opinions (Braylan and Lease, 2020; Li *et al.*, 2021; Weerasooriya *et al.*, 2023) as a way to reduce the annotation bias. Variability in labelling is unavoidable, even if considered on a continuous scale (Sachdeva *et al.*, 2022).

2.2 Modelling hatefulness in other tasks

Methods have also explored using additional tasks to aid hate speech detection, such as emotions (Chiril *et al.*, 2022; Awal *et al.*, 2021) and sentiment (Cao *et al.*, 2020; Zhou *et al.*, 2021*b*), etc. Meanwhile, tasks such as multilabel classification (Ibrohim and Budi, 2019), hate speech diffusion (Masud *et al.*, 2021; Sahnan *et al.*, 2021), rationale and span detection (Vidgen *et al.*, 2021*a*; Mathew *et al.*, 2021), target group prediction (Yoder *et al.*, 2022) have been explored. Building upon the LLM-based data generation of Toxigen (Hartvigsen *et al.*, 2022), adversarial data collection (Ocampo *et al.*, 2023*a*), and LLM-prompting (Kim *et al.*, 2023) have also been explored for improving implicit hate detection. Such adversarial setups require training an additional LLM. *Below, we discuss some tasks that come into use once a speech is detected as hateful. Thus, hate speech detection underpins the work of modelling other hatefulness tasks.*

2.2.1 Generative

Toxicity explanations. One way to help moderators is to incorporate the context by uncovering stereotypical implications. Control over high-level properties of the generated text, such as toxicity, can be obtained by tweaking and promoting certain concepts in the vocabulary space (Geva *et al.*, 2022) of PLMs and LLMs. While understanding toxicity and biases encoded by pretraining data (Ousidhoum *et al.*, 2021) is an essential area of research, our thesis focuses on the downstream finetuning of LMs for hate explanation. Here, post hoc span or scoring explanations techniques are inadequate for implicit spans (Mathew *et al.*, 2021; Masud *et al.*, 2022).

Meanwhile, the work in the form of free text explanations is nascent (Calabrese *et al.*, 2024; Cao *et al.*, 2022; Balkir *et al.*, 2022). It mainly employs variants of LLMs and KG infusion, which may not be feasible to deploy in-house (Sap *et al.*, 2020; ElSherief *et al.*, 2021; Zhou *et al.*, 2023; Mun *et al.*, 2023; Zhang *et al.*, 2023). The most comparable system for us is MIXGEN proposed by Sridhar and Yang (2022). It is a BART-based ensemble of three different knowledge signals (*expert*, *implicit*, *explicit*) that generate implied explanations.

Civility enhancement. In the context of rephrasing offensive text to non-offensive ones, researchers have again explored methods ranging from the rule-based system (29 hand-crafted rules) (Su *et al.*, 2017) to unsupervised text-transfer models (Nogueira dos Santos *et al.*, 2018; Tran *et al.*, 2020). On the line of negativity reduction, practitioners have attempted to increase the politeness of a question/answer (query) pair (Madaan *et al.*, 2020) or reduce subjectivity in news articles (Pryzant *et al.*, 2020). Several studies (both supervised and unsupervised) showed successful rephrasing of a sentence by modifying its sentiment attribute. These methods (Xu *et al.*, 2018; Dai *et al.*, 2019; Shen *et al.*, 2017; Li *et al.*, 2018; Reid and Zhong, 2021; Hu *et al.*, 2017) largely disentangled the sentiment attribute and then relied on a combination of select rephrasing and attention mechanisms to generate an output sequence. Our experiments show that reducing bias or sentiment is inadequate for reducing hate intensity.

In PLMs and LLMs, the toxic tendencies of the systems not only aggregate with finetuning on hateful text but also arise owing to the poisonous and superior correlations (of stereotypes and cultural biases) in the pretraining data (Nadeem *et al.*, 2021). Given the easy-to-use and direct-user-facing prompted interface now being defacto (Morris, 2024), the need for monitoring and controlling toxicity during training (Li *et al.*, 2024; Kim and Lee, 2024) becomes even more necessary.

2.2.2 Diffusive

Spread of hate. Predicting the spread of information on online platforms is crucial in understanding network dynamics, as the destructive power of hate speech lies in its ability to spread across the network. The latest in the family of diffusion is the CHASSIS model (Li *et al.*, 2020; Zhou *et al.*, 2020). On the other end of the spectrum are the susceptibility-infection-recovery SIR (Kermack and McKendrick, 1927) and independent cascade (IC) based models (Bourigault *et al.*, 2016; Gao *et al.*, 2017*b*) adopted from Epidemiology. While highly popular, both DeepCas (Li *et al.*, 2017) and DeepHawkes (Cao *et al.*, 2017) focus only on the size of the overall cascade (macroscopic prediction). For microscopic cascade predictions, recurrent neural architectures (Yang *et al.*, 2018), such as TopoLSTM (Wang *et al.*, 2017), FOREST (Yang *et al.*, 2019) have been explored, which only consider network-level signals (who is posting and sharing), and not the content of the post (what is being shared). In this regard, TopoLSTM considers only the previously seen nodes in any cascade as the next candidate without using timestamps as a feature. The approximation works well under the limited availability of network information and the absence of cascade metadata. Meanwhile, FOREST considers all the users in the global graph (irrespective of one-hop) as potential users, employing a time-window-based approach.

The spread of hate and its exploratory analysis (Mathew *et al.*, 2019; Ribeiro *et al.*, 2018) revealed exciting characteristics of the breadth and depth of hate vs non-hate diffusions when examined separately. Real-world interactions are more convoluted, with the same communication thread containing hateful, counter-hateful, and non-hateful comments (Sahnan *et al.*, 2021). Thus, independent diffusion studies, while adequate for the exploratory analysis of hate, cannot be directly extrapolated for predictive analysis of hate diffusion.

2.3 Infusing context

From employing world news data for enhancing LMs (Wenzek *et al.*, 2020) to boosting the impact of online advertisement campaigns (Muhlmeyer *et al.*, 2019) and virality (Myers *et al.*, 2012), exogenous influence has been successfully applied in a wide variety of tasks. Especially in terms of opinion and chatter prediction as well as user engagement of multimedia content, the superiority of models that consider exogenous signals has been adequately established (De *et al.*, 2018; Dutta *et al.*, 2020; Hu *et al.*, 2015; Khosla *et al.*, 2014).

2.3.1 Linguistic features

To determine the hateful text, most of these models utilise a static-lexicon-based approach and consider each post/comment in isolation. With a lack of context (both in the form of an individual's prior indulgence in the offence and the current worldview), the models trained on previous trends perform poorly on new datasets (Kulkarni *et al.*, 2023). For explicit hate speech, the role of external hate lexicons as one-hot encoded vectors (Polignano *et al.*, 2022; Stamou *et al.*, 2022; Gao *et al.*, 2017*a*) has been explored. Another direction is the use of joint learning to capture both emotion and hate labels, where the additional information from negative emotions can be employed as a latent signal to improve hate detection (Awal *et al.*, 2021; Schäfer and Kistner, 2023). Meanwhile, efforts are being made to examine the usage of topical, historical (Qian *et al.*, 2018) context for hate speech. Conversation thread/timeline can also be employed to improve the detection of hate and capture implicitness in long-range contexts (Ghosh *et al.*, 2023).

2.3.2 Topological features

External events such as pandemic (He *et al.*, 2022), activism (Massanari, 2017; Chatzakou *et al.*, 2017), elections (Masud and Chakraborty, 2023; LAI *et al.*, 2023), etc., can prompt the social network to engage in more hateful content. Only recently have researchers started using network-level information for hate speech detection (Ghosh Chowdhury *et al.*, 2019; Fehn Unsvåg and Gambäck, 2018).

2.3.3 Knowledge tuples

KGs (Speer and Havasi, 2012; Sap *et al.*, 2018) are often applied in NLP (Schneider *et al.*, 2022; Pan *et al.*, 2024; Yu *et al.*, 2022) for reasoning (Chang *et al.*, 2020), question answering (Feng *et al.*, 2020), story generation (Guan *et al.*, 2020), sarcasm explanation (Kumar *et al.*, 2022), etc. The role of world knowledge has also been explored for hate target detection (Reyero Lobo *et al.*, 2023), and implicit type classification (ElSherief *et al.*, 2021; Lin, 2022). Meanwhile, Deshpande *et al.* (2022) released a stereotype-focused KG targeting six nationalities and religions.

Part II

Establishing the importance of contextual signals

CHAPTER 3

Contextualised Hate Prediction

"Words can be like tiny doses of arsenic: they are swallowed unnoticed, appear to have no effect, and then, after a little time, the toxic reaction sets in after all."
- Viktor Klemperer; The Language of the Third Reich

3.1 Chapter introduction

By design, OSNs have democratised the power of content generation in the hands of every user of these platforms (Munn, 2020). Bad agents exploit this design to disseminate hate campaigns (among other harmful content) to a degree where manual monitoring is no longer feasible. An early prediction of how hateful content is propagating can help in combating it. It is, however, important to reiterate that despite a visible increase in hatefulness, toxic content still forms a small proportion of the overall content on the Web. The setting makes the application of contextual signals even more interesting, with the question, "*To what extent can contextual richness make up for class imbalance?*"

We begin exploring the above theme by empirically establishing the role of contextual information in the diffusion dynamics of retweeting hateful content on X. The task is accomplished via curation of the novel context-rich English dataset of Indian topics via the `ConInHate` (**C**ontextual **I**ndian **H**ate) dataset and a feature-engineered machine learning framework. `RETINA` (**Ret**weeter **I**dentifier **N**etwork with Exogenous **A**ttention) employs a combination of endogenous and exogenous signals to enhance the retweet prediction system. An in-depth comparison with existing information diffusion baselines reveals that context-rich models, like `RETINA` can help learn patterns of hateful spread even under skewed class proportions. The use of a feature-engineered hate speech detection system in `RETINA` coincides with the rising popularity of BERT-based PLMs. Yet, there is little research into how various critical aspects of PLMs affect their performance in hate speech detection. As the first step towards employing PLMs for hate speech detection, a thorough examination of English-only datasets and popular PLMs is conducted. The findings from PLM finetuning and modelling contextual information for English content contribute towards developing contextual hate speech detection models in a multilingual setting. We assess this by proposing the `HEN-mBERT` (**H**istory-**E**xamplar-**N**etwork **I**nfused m**B**ERT) model. Its efficacy is judged by curating a context-rich Hinglish dataset from X called `GOTHate` (**G**eo-p**O**litical **T**opical **H**ate dataset). For multilingual hate speech detection, the incoming post is augmented with the user's timeline information and ego network, bringing the setup closer to real-time human moderation that has access to such platform-specific signals.

Working with diffusion models and topological features, we observe a vacuum among easy-to-use interfaces for large-scale visualisation of diffusion models. We end this chapter by introducing `DiVA` (**Di**ffusion **V**isualisation and **A**nalysis), a tool that provides a scalable web interface and extendable APIs to analyse various diffusion trends on a network.

3.2 A topic-aware modelling of hate speech diffusion

3.2.1 Motivation

Research gap. In order to gain a better understanding of how hate spreads across a network, initial works (Ribeiro *et al.*, 2018; Mathew *et al.*, 2019) have attempted to provide an exploratory analysis. However, their methodology separates the non-haters from haters and studies the diffusion of two cascades independently. The setup restricts their ability to be directly extrapolated for predictive analysis of hateful posts. Meanwhile, as early as 2012, studies (Myers *et al.*, 2012) have exposed that external stimuli drive one-third of the information diffusion on Twitter, an analysis of which in relation to hate speech is underexplored. Moreover, the studies that employ social media data for information diffusion either focus on the size of the cascade (Li *et al.*, 2017; Cao *et al.*, 2017) or only look at network information (Li *et al.*, 2020; Wang *et al.*, 2017). By not accounting for textual context, the nature and extent of hatefulness are not captured. These research gaps necessitate the introduction of contextual information that can act as stimuli for analysing and predicting the spread of online hate speech.

Research questions. To overcome the above research gaps, this study seeks to answer the following research questions (RQs):

RQ1: To what extent do endogenous signals help analyse the spread of hateful posts?
RQ2: How can endogenous and exogenous signals be effectively combined to predict the spread of hateful posts?

Contribution summary. In order to explore the user behaviour that triggers diffusion of hate speech via retweets, we crawl a large-scale dataset of tweets, retweets, user activity history, and follower networks based on trending hashtags on X. The dataset is henceforth referred to as `ConInHate`. It consists of $\approx$ 30k unique root tweets and 13k unique root users. Since the data is collected based on trending Indian hashtags, it becomes crucial to model exogenous signals, some of which may have triggered the trend in the first place. While a one-to-one mapping of news keywords to trending keywords is challenging to obtain, we collate the most recent (time window) news with respect to a source tweet as our ground truth. *To our knowledge, this is the first retweet prediction dataset to include endogenous and exogenous influences (Masud* et al.*, 2021).*

Compared to existing models for microscopic cascade prediction, which aim to answer who will be the next participant in the cascade, our work aims to determine whether a follower of a user will retweet a post. This converts our use case into a binary classification problem. *Adding negative sampling (in the form of inactive nodes) takes the proposed setup closer to a real-world scenario consisting of active and passive social media users.* Modelling the `ConInHate` dataset, we propose `RETINA`, a neural architecture to predict hateful retweeters. `RETINA` significantly outperforms several state-of-the-art information diffusion models, achieving a macro F1-score of 0.85 (Masud *et al.*, 2021).

3.2.2 Dataset curation

To overcome the lack of context-rich hate-inducing datasets for Indian topics, we curate `ConInHate` from X (Twitter). Apart from the source post/tweet, `ConInHate` contains an array of textual, temporal, and topological information. Using Twitter's official API (available for free in 2020), we track the Indian socio-political space for trending hashtags each day from 03/02/2020 to 14/04/2020. From 34 unique hashtags, we obtain $31,133$ tweets from $13,965$ users. These tweets are hereby referred to as source/root tweets. Note that the content is filtered to retain only those whose detected language is English. We then crawl the list of retweeters per source post. The information network is built by crawling the followers of each unique user in the datasets, along with their activity history/timeline.

External stimuli. Using the news-please crawler (Hamborg *et al.*, 2017), we crawl online news articles published within the period from 03/02/2020 to 14/04/2020. We collect a total of $683,419$ news articles. After filtering for language, title, and date, we are left with $319,179$ news items in English.

3.2.3 Dataset annotation

Annotator details. We employ three professional annotators who have experience in analysing online hate speech to annotate the source tweets manually. The annotators aged between 22-27 are active Twitter users. As the contextual knowledge of real-world events plays a crucial role in identifying hate speech, we ensure that the annotators are well aware of the socio-political events under consideration.

Annotation process. To begin with, each annotator accesses 100 random source posts and annotates them as hateful or not based on their world knowledge. An online workshop is conducted with the annotators to discuss their findings and challenges, after which the guideline is narrowed to mimic X's content moderation policy as of April 2020[1]. Under this setup, we obtain binary annotations (hate or non-hate) for $23,748$ tweets with an (inter-annotator agreement) IAA of 0.58 Krippendorf's α. The low agreement scores point towards the difficulty of the task at hand. The final labels for a tweet are assigned based on majority voting, leading to $1.1k$ root tweets labelled as hateful.

Labelling for the rest of the data. For source tweets which are not manually annotated, as well as the tweets from users' timelines, we develop a pseudo-labelling system. We train three different feature-engineered hate speech classifiers based on the designs proposed by Davidson Davidson *et al.* (2017) (dubbed as the Davidson model), Waseem Waseem and Hovy (2016), and Pinkesh Badjatiya *et al.* (2017). With a receiver operating characteristics area under the curve (ROC-AUC) of 0.85 and macro-F1 0.59, the Davidson model is the best performing, as recorded in Table 3.1. The results are obtained on $k = 3$ cross-validation training over the $23k$ annotated samples. We label the rest of the tweets in the dataset via the predictions of our custom Davidson model. Note, when using the off-the-shelf Davidson model, which had been trained on its own English dataset, we registered test scores of 0.79 AUC and 0.48 macro-F1.

[1] `https://help.twitter.com/en/rules-and-policies/hateful-conduct-policy`

Table 3.1: Performance of hate speech detection models trained on annotated samples of `ConInHate`.

Classifier	**Macro-F1**	**ROC-AUC**
`Davidson`	0.59	0.85
`Waseem`	0.50	0.87
`Pinkesh`	0.49	0.50

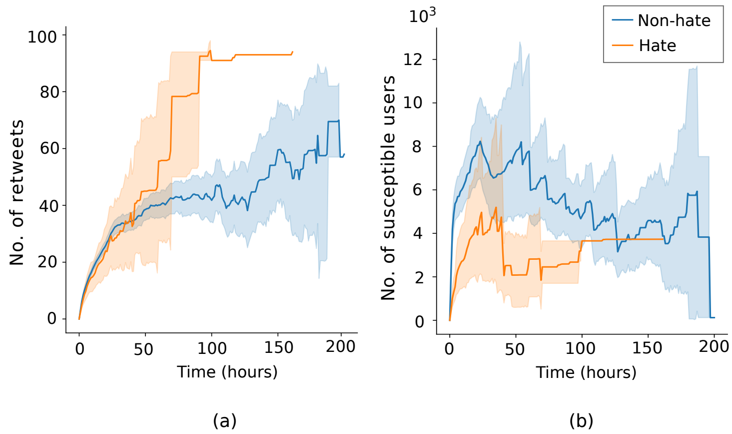


Figure 3.1: Plot (a) shows the growth of retweet cascades for hateful and non-hateful tweets (solid lines and shaded regions signify the average over the dataset and confidence of count, respectively). Analogously, plot (b) depicts the temporal change of susceptible users over time.

> The performance gap of 0.79 vs 0.85 AUC between off-the-shelf and finetuned Davidson models highlights the generalisability limits of Western-centric hate detection models in capturing the Indian context, even when both sets of tweets are in the English language.

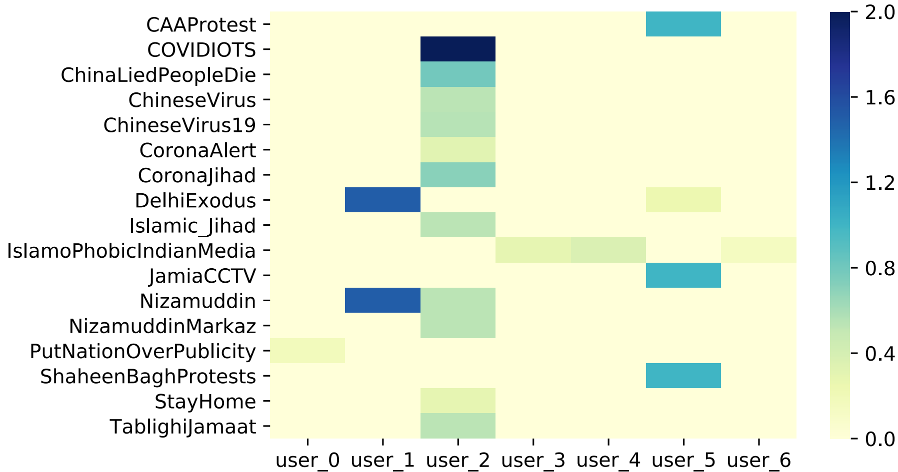


Figure 3.2: Distribution of hateful to non-hateful tweets by a random set of users.

3.2.4 Analysing diffusion patterns of hate vs non-hate

Rate of diffusion. We begin our assessment by looking at the temporal dynamics of hateful vs non-hateful source tweets. Following the standard information diffusion

Table 3.2: Statistics of `ConInHate`: *Ave. RT*, *Users*, and *Users-all* signify average retweets, the unique number of users tweeting, and the unique number of users engaged in (tweet+retweet) the #-tag, respectively. JV: *jamiaviolence*, MOTR: *MigrantsOnTheRoad*, TTSV: *timetosackvadras*, JUA: *jamiaunderattack*, IBN: *IndiaBoycottsNPR*, ZNBK: *ZeeNewsBanKaro*, SCW: *SaluteCoronaWarriors*, IPIM: *IslamoPhobicIndianMedia*, DR2020: *delhiriots2020*, S4S: *Seva4Society*, PMCF: *PMCaresFunds*, C_19: *COVID_19*, HUA: *Hindus_Under_Attack*, WP: *WarisPathan*, LE: *lockdownextension*, JCCTV: *JamiaCCTV*, TVI: *TrumpVisitIndia*, PNOP: *PutNationOverPublicity*, DE: *DelhiExodus*, DER: *DelhiElectionResults*, ASMR: *amitshahmustresign*, R4GK: *Restore4GinKashmir*, DV: *DelhiViolance*, SNPR: *StopNPR*, 1C4DH: *1Crore4DelhiHindu*, NV: *NirbhayaVerdict*, NM: *NizamuddinMarkaz*, 90DSB: *90daysofshaheenbagh*, DEM: *Demonetisation*, NHR: *NorthDelhiRiots*, PMP: *PMPanuti*, HLM: *HinduLivesMatter*, CV: *ChineseVirus*, UM: *UmarKhalid*.

#-tags	JV	MOTR	TTSV	JUA	IBN	ZNBK	SCW	DEM	CV
Tweets	950	872	280	263	570	919	104	1696	8
Avg. RT	15.45	6.69	8.19	5.8	7.87	9.58	5.65	3.46	0.25
Users	743	641	138	215	333	751	53	607	7
Users-all	4026	2176	548	688	1227	1940	225	4494	8
%-Hate	3.78%	8.20%	1.3%	6.06%	0.8%	7.01%	0.0%	0.06%	0.5%
#-tags	IPIM	DR2020	S4S	PMCF	C_19	HUA	WP	NHR	UM
Tweets	4307	1453	1087	1172	971	382	989	3418	887
Avg. RT	15.46	12.23	13.24	7.61	6.38	7.10	9.23	2.89	3.82
Users	1181	1136	532	1076	807	292	807	1316	439
Users-all	3237	6051	4058	2691	2593	1073	2924	7251	2510
%-Hate	8.42%	6.8%	1.53%	0.8%	1.96%	10.1%	12.07%	0.08%	0.1%
#-tags	LE	JCCTV	TVI	PNOP	DE	DER	ASMR	PMP	–
Tweets	107	1045	339	555	542	843	959	1346	–
Avg. RT	1.85	12.07	8.47	13.24	9.66	7.56	5.01	4.06	–
Users	102	815	284	365	414	731	765	368	–
Users-all	138	4091	1134	2146	1857	1807	1807	2310	–
%-Hate	0.0%	5.66%	2.6%	5.71%	7.61%	3.20%	9.94%	0.02%	–
#-tags	R4GK	DV	SNPR	1C4DH	NV	NM	90DSB	HML	–
Tweets	949	1121	82	889	649	1124	226	392	–
Avg. RT	3.94	9.004	10.23	11.62	7.61	8.24	5.25	4.82	–
Users	492	948	64	770	546	843	188	145	–
Users-all	986	2702	440	3045	1577	3199	506	1396	–
%-Hate	2.84%	7.37%	0.0%	0.99%	4.67%	7.85%	12.04%	0.12%	–

terminology, the set of susceptible user nodes at any time is the set of all the nodes that have been exposed to the information at a given timestamp (followers of those who have posted/retweeted). Note that susceptibility is defined by exposure but not engagement. Hence, a susceptible user may engage with the content in the future, but currently has not (did not retweet/like/comment). Figure 3.1(a) highlights that hateful tweets are quickly retweeted in a higher magnitude compared to non-hateful ones. Moreover, Figure 3.1(b) showcases that hateful tweets acquire most of their retweets and susceptible nodes in a very short time. A possible explanation for these observations is the echo chamber effect. Hate speech is often characterised by the formation of echo chambers, i.e., only a small group of people engaging with similar content repeatedly (Goel *et al.*, 2023).

Topical affinity. Table 3.2 presents a granular analysis of `ConInHate` based on a combination of manual and pseudo labelling. As evident from Figures 3.2 and 3.3, the degree of hatefulness expressed by a user is dependent on the topic as well. Even when different hashtags share a common theme, they still incur a different degree of

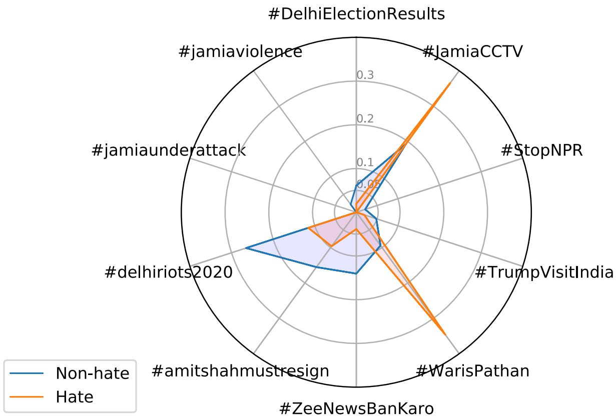


Figure 3.3: Distribution of hateful vs non-hateful tweets on a scale of 0 to 1 for all the users for a selected number of hashtags.

engagement in terms of source tweets and hatefulness. As is the case with #jamiaunderattack (JUA), #jamiaviolence (JV), and #jamiaCCTV (JCCTV). On topics with common themes that have a similar number of source tweets, the level of hatefulness still varies. As is the case with #HindusUnderAttack (HUA) and #HinduLivesMatter (HLM). Within India's context, considering the first wave of COVID-19, we see social topics such as #MigrantsOnTheRoad (MOTR) incur higher-than-expected hatefulness.

3.2.5 Design of `RETINA` for retweet prediction

Inspired by our observation of the diffusion patterns (Section 3.2.4), we formulate the problem of diffusion prediction and enlist the features employed to model the same.

Problem statement. Given a tweet τ posted by some root user by u_0 at timestamp t_0, we formulate the problem as predicting the potential retweeters within the interval $[t_0, t_0 + \Delta t]$, from among the $u_j \in U$ users in the dataset. We account for contextual signals from the history of activity $\mathcal{H}_{j,t}$ of u_j up to timestamp t and the peer influence $\mathcal{S}_j^P$ of u_i on u_j. Further accounting for interaction being influenced by trending hashtags and world news, we have two more endogenous and exogenous influences captured in $\mathcal{S}_j^{\text{en}}$ and $\mathcal{S}_i^{\text{ex}}$, respectively.

Now, assuming the probability density of a user u_j retweeting τ at time t to be $p(t)$, then retweet prediction problem translates to learning the parametric function $f : \mathbb{R}^d \to (0, 1)$ such that:

$$\int_{t_0}^{t_0+\Delta t} p(t)dt = f(\mathcal{S}_j^P, \mathcal{S}_j^{\text{en}}, \mathcal{S}_i^{\text{ex}}, \mathcal{H}_{j,t}, \tau|\theta) \tag{3.1}$$

However, t_i is not the time of posting of the u_i user; the two notations are not attached via the subscript; one denotes the ith user in the network, and the other is the ith timestamp. In Equation 3.1, θ is the learnable parameter.

We expand below on how each feature in Equation 3.1 is obtained and trained.

Peer influence. For a user u_j, we incorporate $\mathcal{S}_j^P$ using two different features. We compute the shortest path length from u_0 to u_j in $\mathcal{G}$. Here $\mathcal{G}$ denotes the user network we curate from Twitter. The first short path captures how quickly u_j can be exposed to τ. We also include the number of times u_j has retweeted tweets by u_0 in the past, signifying alignment of ideology.

Historical influence. The activity history of user u_j, signified by $\mathcal{H}_{j,t}$, is available in the form of the content they have posted on their timeline. We record historical influence on their current retweeting ability via the following:

- Follower count and date of account creation of u_j.
- Number of unique topics (hashtags) u_i has tweeted on up to t.
- We use unigram and bigram features weighted by TF-IDF values from 30 most recent tweets posted by u_j to capture their recent topical interest. To reduce the feature space, only the top 300 features sorted by their IDF values are considered.
- To capture the history of hate generation by u_j, we compute three different features on their 30 most recent tweets:
  - The ratio of hateful vs. non-hateful tweets of u_j up till time t.
  - The ratio of hateful vs. non-hateful retweets of u_j. The higher engagement a user receives on their hateful post, the more likely they are to engage and support other hateful content.
  - It is a count/frequency vector of length 209 from among the tweet history mapped against the 209 Indian hate slur terms borrowed from Kapoor *et al.* (2018). Examples of slur terms used in the lexicon include words such as *ha***i* (bastard), *jh*ll** (faggot), etc. Using the above terms is derogatory and a direct offence. In addition, the lexicon has some colloquial terms such as *mulla* (Muslim), *bakar* (gossip), *aktakvadi* (terrorist), *jamai* (son-in-law), *haathi* (elephant/fat), which may carry a hateful sentiment depending on the context in which they are used.

Note that we did experiment with variable feature lengths, but TF-IDF at 300 and 30 for the timeline worked best.

Non-peer endogenous feature. We incorporate $\mathcal{S}^{\text{en}}$ by supplying the model with a binary vector representing the top 50 trending hashtags for the day the tweet is posted.

Endogenous representation. The vector representation of a user u_j is obtained as a concatenation of (:) of their endogenous influences as outlined in Equation 3.2.

$$\mathbf{X^{u_j}} = [\mathcal{S}_j^P : \mathcal{S}_j^{\text{en}} : \mathcal{H}_{j,t}] \tag{3.2}$$

Exogenous feature. For the retweet prediction task, we incorporate the exogenous signal using two different methods. To implement the attention mechanism of `RETINA`, we use a doc2vec representation of the n most recent news headlines from our corpus posted before the time of the tweet. Our best results were obtained with $n = 60$. For the baseline models, we compute the average TF-IDF vector for the $n = 60$ most recent news headlines from our corpus posted before the time of the tweet. Again, we select the top 300 features.

Exogenous attention. To incorporate external information as an assisting signal to model diffusion, we employ the cross-attention variant of the scale dot product (Vaswani *et al.*, 2017). The vanilla self-attention mechanism has the input for query, key, and value coming from the same vector. In our case, given a source tweet (query), we want to map the influence of a set of n news headlines that occurred around the same time the tweet

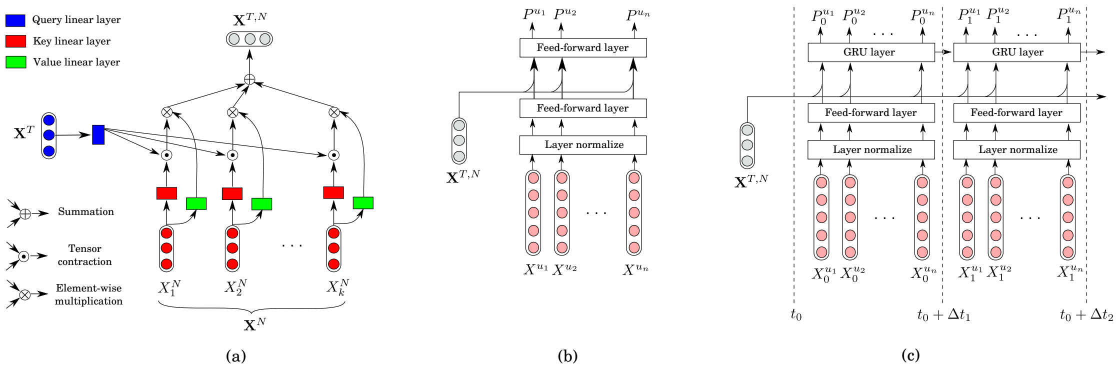


Figure 3.4: Design of different components of `RETINA`: (a) *Exogenous attention*: Key and Value linear layers (blue) are applied on each element of the news feature sequence $\mathbf{X}^N$, while the Query linear layer (red) is applied on the tweet feature $\mathbf{X}^T$. The attention weights computed for each news feature vector by contracting the query and key tensors along the feature axis (dot product) are then applied to the value tensors and summed over the sequence axis to produce the 'attended' output, $\mathbf{X}^{T,N}$. (b) *Static prediction of retweeters*: To predict whether u_j will retweet, the input feature X^{u_j} is normalised and passed through a feed-forward layer, concatenated with $\mathbf{X}^{T,N}$, and another feed-forward layer is applied to predict the retweeting probability P^{u_j}. (c) *Dynamic retweet prediction*: In this case, `RETINA` predicts the user retweet probability for consecutive time intervals, and instead of the last feed-forward layer used in the static prediction, we use a GRU layer.

was posted. Hence, for the incoming tweet query, the key and value stores are obtained from the news vectors. Given the doc2vec feature representation of the tweet $\mathbf{X}^T$ and a n set of news headlines represented as Equation 3.3.

$$\mathbf{X}^N = \{X_1^N, X_2^N, \dots, X_n^N\} \tag{3.3}$$

We compute the query ($\mathbf{Q}^T$), key ($\mathbf{K}^N$) and value ($\mathbf{V}^N$) tensors via Equation 3.4. Here $\mathbf{W}^Q$, $\mathbf{W}^K$, and $\mathbf{W}^V$ are learnable parameters, and $\mathbf{A^{T,N}}$ is the attention weight between the tweet and news sequence. Each of $\mathbf{W}^Q$, $\mathbf{W}^K$, and $\mathbf{W}^V$ is a two-dimensional tensor with $hdim$ columns. The attention weight is then used to produce the final encoder feature representation $\mathbf{X}^{T,N}$ as a weighted average of $\mathbf{V}^N$.

$$\begin{aligned}\mathbf{Q}^T = \mathbf{X}^T \odot \mathbf{W}^Q, \mathbf{K}^N = \mathbf{X}^N \odot \mathbf{W}^K, \mathbf{V}^N = \mathbf{X}^N \odot \mathbf{W}^V \\ \mathbf{A^{T,N}} = Softmax(\frac{\mathbf{Q}^T \odot \mathbf{K}^N}{\sqrt{hdim}}) \\ \mathbf{X}^{T,N} = \mathbf{V}^N \odot \mathbf{A^{T,N}}\end{aligned} \tag{3.4}$$

Final retweet prediction. `RETINA` aggregates the exogenous signal exposed by the sequence of news inputs according to the feature representation of the tweet into $\mathbf{X}^{T,N}$, using the operations mentioned in Equations 3.4 via tuning the $\mathbf{W}^*$ parameters. The setup is summarised in Figure 3.4 (a). To perform the task of retweet prediction, we integrate the exogenous-influenced source tweet ($\mathbf{X}^{T,N}$ with the endogenously influenced user representation ($\mathbf{X^{u_j}}$ from Equation 3.2) to predict the retweeting probability of u_j. Here, we introduce two modes of retweet prediction:

- **Static prediction.** Here, t_0 is fixed, and Δt is ∞, i.e., all the retweeters, irrespective of their retweet time (no temporal ordering, Figure 3.4 (b)). The feature vector X^{u_j} corresponding to user u_j is first normalised and mapped to an intermediate representation using an FNN layer. It is then concatenated with the output of the exogenous attention component, $\mathbf{X}^{T,N}$. Finally, another feed-forward layer with sigmoid nonlinearity is applied to compute the probability P^{u_j}.
- **Dynamic prediction.** To incorporate the temporal dynamics as observed early in our diffusion analysis (Figure 3.1), we develop the dynamic setting. In the dynamic setting `RETINA` predicts the probability of every user u_j to retweet within a time interval $[t_0 + \Delta t_i, t_0 + \Delta t_{i+1}]$,... and so on, with t_0 being the time of the tweet published and $\Delta t_0 = 0$. To capture the temporal dependency between predictions in successive intervals, we replace the last feed-forward layer with a Gated Recurrent Unit (GRU), as shown in Figure 3.4 (c). We experimented with other recurrent architectures as well; performance degraded with simple RNN and no gain with LSTM. The GRU layer latently introduces the decay effect, where we hope to capture a reduced influence of the tweet on those who interact with it later.

Cost/loss function. In both settings (static and dynamic), the task translates to a binary classification problem. Therefore, we use standard binary cross-entropy (CE) loss L to train `RETINA` via $L = -w \cdot g \log(p) - (1 - g) \log(1 - p)$. In the overall loss, g is the ground truth, p is the predicted probability, and w is a weight given to the positive samples to deal with class imbalance.

3.2.6 Experimental setup

Data splits. Starting with a corpus of $31,133$ source tweets, on average, a source tweet is retweeted 13.10 times, with only one tweet being retweeted 196 times. After filtering only those tweets that have more than one retweet and at least 60 news mappings from the time of their posting, we are left with 3822 source tweets. With an $80:20$ train-test split, this resulted in a total of 3057 and 765 cascade samples for training and testing.

Hyperparameters. For both static and dynamic prediction of retweeters, we use mini-batch training of `RETINA`. We vary the batch size within 16, 32, and 64, with the best results on a batch size of 16 for the static mode and 32 for the dynamic mode. We also vary the learning rates within a range of 10^{-4} to 10^{-1} and choose the best one with a learning rate of 10^{-2} using the SGD optimiser for the dynamic model. The static counterpart produced the best results with the Adam optimiser using default parameters. To deal with the class imbalance, we set the parameter w in loss as $w = \lambda(\log C - \log C^{+})$, where C and C^{+} are the counts for total and positive samples, respectively in the training dataset, and λ is a balancing constant which we vary from 1 to 2.5 with 0.5 steps. We find the best configurations with $\lambda = 2.0$ and $\lambda = 2.5$ for the static and dynamic modes, respectively. For all the doc2vec-generated feature vectors related to tweets and news headlines, we set the dimensionality to 50 and 500, respectively. For `RETINA`, we set the parameter $hdim$ and all the intermediate hidden sizes for the feed-forward and recurrent layers to 64.

Evaluation metrics. We report macro-F1, AUC, and binary accuracy (ACC). As the neural baselines tackle the problem of retweet prediction as a ranking task, we improvise the evaluation of `RETINA` to make it comparable with these baselines. We rank the predicted probability scores to compute mean average precision (MAP@k) and binary hits at top-k positions (HITS@k).

Baselines. We implement 9 external baselines for comparison. Since information diffusion is a vast subject, we approach it from three perspectives – one is the set of rudimentary diffusion baselines, one is a group of traditional ML models, and the other is the set of neural models.

- **SIR.** The Susceptible-Infectious-Recovered/Removed (Kermack and McKendrick, 1927) setup is one of the earliest predictive models for the spread of information adopted from Epidemiology. Two parameters govern the SIR model – transmission rate and recovery rate, which dictate the spread of contagion (retweeting in our case) in the network.
- **Threshold model.** Another traditional system, the threshold model (Kempe *et al.*, 2003), assumes that each node has threshold inertia chosen uniformly at random from the interval $[0, 1]$. A node becomes active (i.e., a retweeter) if the weighted sum of its active neighbours exceeds this threshold.
- **Feature engineering-based systems.** Using the same endogenous and exogenous feature set as described in Equations 3.2 and 3.3, we employ four classifiers – Logistic Regression, Decision Tree, Linear SVC, and Random Forest (with 50 estimators). All of these models are used only for the static mode of retweet prediction without the attention mechanism. The exogenous and endogenous features are concatenated.
- **TopoLSTM.** TopoLSTM (Wang *et al.*, 2017) is one of the initial works to consider recurrent models in generating user prediction probabilities. The model converts the cascades into dynamic directed acyclic graphs (DAGs) to capture the temporal signals via node ordering. The sender-receiver-based RNN model captures a combination of the active node's static score (based on the existing state of the cascade) and a dynamic score (capturing future propagation tendencies). Note that both these scores only capture the network feature.
- **FOREST.** Proposed as a unified model to perform the microscopic and the macroscopic cascade predictions, FOREST (Yang *et al.*, 2019) combines reinforcement learning (for macroscopic) with the recurrent model (for microscopic). By considering the global graph, graph sampling is performed to obtain the structural context of a node as an aggregate of the structural context of its one or two-hop neighbours. In addition, it factors the temporal information via the last m seen nodes in the cascade.
- **HIDAN.** To reduce memory overhead, HIDAN (Wang and Li, 2019) does not explicitly consider a global graph as input. Any information loss due to the absence of a global graph is substituted by temporal information utilised in the form of the ordered time difference of node infection. It considers the set of all seen nodes in the cascade as candidate nodes for prediction.

Ablations. With the exogenous attention component removed in static (`RETINA`-S†) as well as dynamic (`RETINA`-D†), the attention mechanism gets replaced by standard self-attention. Further, we remove the exogenous feature in the traditional baselines.

Inspired by the use of contextual signals, we perform additional experiments to determine if these signals can help predict which users are more likely to generate hateful content. The extended experiments and results are available in Appendix A.

Table 3.3: Performance of RETINA and other baselines for retweeter prediction. RETINA-S and RETINA-D correspond to static and dynamic prediction settings, respectively. Gen.Thresh. corresponds to the General Threshold model. † symbolizes models without exogenous signal.

Model	**Macro-F1**	**ACC**	**AUC**	**MAP@20**	**HITS@20**
Logistic Regression	0.70	0.96	0.79	-	-
Logistic Regression†	0.49	0.93	0.50	-	-
Decision Tree	0.68	0.95	0.78	-	-
Decision Tree†	0.54	0.92	0.54	-	-
Random Forest	0.66	0.97	0.67	-	-
Random Forest†	0.52	0.93	0.52	-	-
Linear SVC†	0.49	0.91	0.50	-	-
RETINA-S	0.70	0.97	0.73	0.57	0.74
RETINA-S†	0.65	0.93	0.74	0.56	0.76
RETINA-D	**0.89**	0.99	**0.86**	**0.78**	**0.88**
RETINA-D†	0.87	0.99	0.80	0.69	0.80
FOREST	-	-	-	0.51	0.64
HIDAN	-	-	-	0.05	0.05
TopoLSTM	-	-	-	0.60	0.83
SIR	0.04	-	-	-	-
Gen.Thresh.	0.04	-	-	-	-

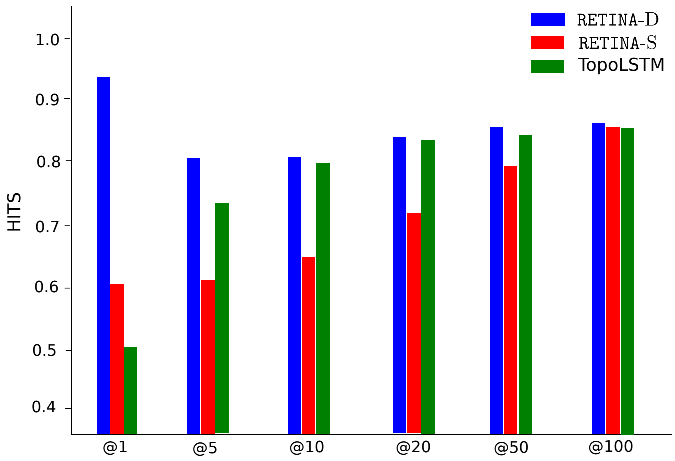


Figure 3.5: Comparing RETINA-D, RETINA-S, and TopoLSTM at HITS@k $=$ $1, 5, 10, 20, 50, 100$.

3.2.7 Results

Overall assessment. Table 3.3 summarises the performances of the competing models for the retweet prediction task. While RETINA in the dynamic setting outperformed the rest of the models by a significant margin under all the evaluation metrics, TopoLSTM emerges as a competitive baseline in terms of both MAP@20 and HITS@20. Among the traditional baselines, Logistic Regression gives a comparable macro-F1 to the best static model; however, owing to memory limitations, it cannot be trained on the news set larger than $n = 15$ per tweet. Similarly, SVM-based models cannot incorporate even $n = 15$ news items per tweet (memory limitation). Meanwhile, an ablation on news size gives the best results at $n = 60$ for both static and dynamic models. Across all setups, including ours, where the exogenous signal is removed, performance drops by a significant margin. The impact of exogenous signals on macro-F1 is more visible in the traditional baselines. Between the static and dynamic RETINA variants, the performance drop is more in RETINA-D† in terms of MAP@k and HITS@k.

Revisiting our best baseline TopoLSTM (HITS@20 of 0.83), which works solely on network information (who follows whom), we further compare RETINA in the static and dynamic setting with TopoLSTM in terms of HITS@k for different values of k. From Figure 3.5, it is evident that as k increases, the three models converge to similar performance. However, the predictive power of RETINA lies in its ability to work effectively for smaller values of k as well. For smaller values of k, RETINA vastly outperforms TopoLSTM in both dynamic and static settings.

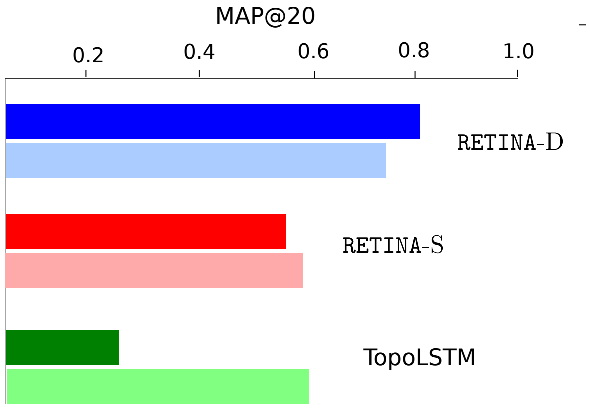


Figure 3.6: Comparison of static (red; RETINA-S) and dynamic (blue; RETINA-D) settings with TopoLSTM (green) to predict retweeters when the root tweet is – hateful (dark shade) vs non-hate (lighter shade).

To provide more holistic insights regarding the retweet diffusion power of our proposed framework, we further examine TopoLSTM against RETINA in terms of hateful and non-hateful test cascades. Figure 3.6 showcases that TopoLSTM fails to capture the different diffusion dynamics of hate speech in contrast to non-hate (MAP@20 0.59 for non-hate vs 0.43 for hate). On the other hand, RETINA achieves MAP@20 scores 0.80 and 0.74 in dynamic (0.54 and 0.56 in static) settings to predict the retweet dynamics for hate and non-hate contents, respectively. This smaller difference can be attributed to both extensive feature engineering as well as employing weighted class loss in training RETINA. Finally, even when not explicitly mapped 1-1 to tweet content, the exogenously informed tweet is better placed to capture the diffusion of hateful tweets.

Given that there are fewer hateful cascades to learn from, TopoLSTM is not able to capture the patterns of hateful diffusion via network semantics alone. Incorporating hate via endogenous and exogenous features enriches RETINA with superior expressive power. It is akin to the cold start problem in recommendation systems, where external knowledge helps bridge the information gap.

Impact of temporal intervals. To observe the performance of RETINA more closely in the dynamic setting, we analyse its performance over successive prediction intervals. Figure 3.7 shows the ratio between the predicted and the actual number of retweets at different intervals. Akin to any information diffusion system, RETINA-D tends to be nearly perfect in predicting new growth with increasing time. The high error rate at the initial stage is due to the fact that the retweet dynamics remain uncertain at first and become more predictable as more people participate over time.

Impact of cascade size. In the absence of temporal intervals, i.e., recording the test metrics based on the number of actual retweets. Akin to RETINA-D, Figure 3.8(a)

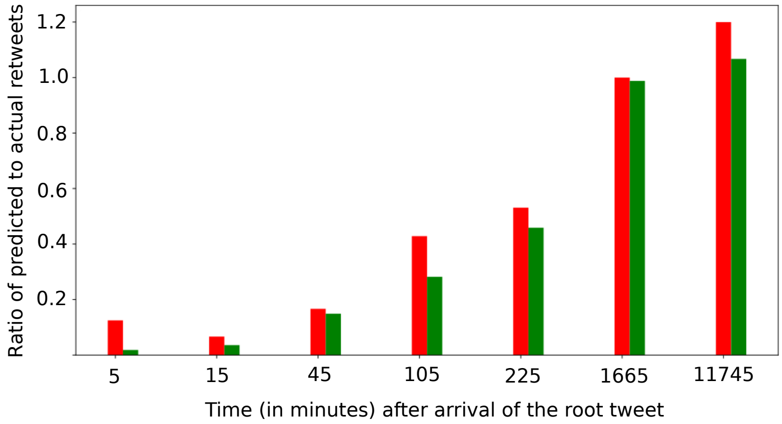


Figure 3.7: Ratio of the number of predicted to actual retweets arrived within different time windows after the arrival of the root tweet for RETINA-D. Red and green bars correspond to hateful vs non-hateful root tweets. Counts of actual and predicted retweets are taken between each successive time interval.

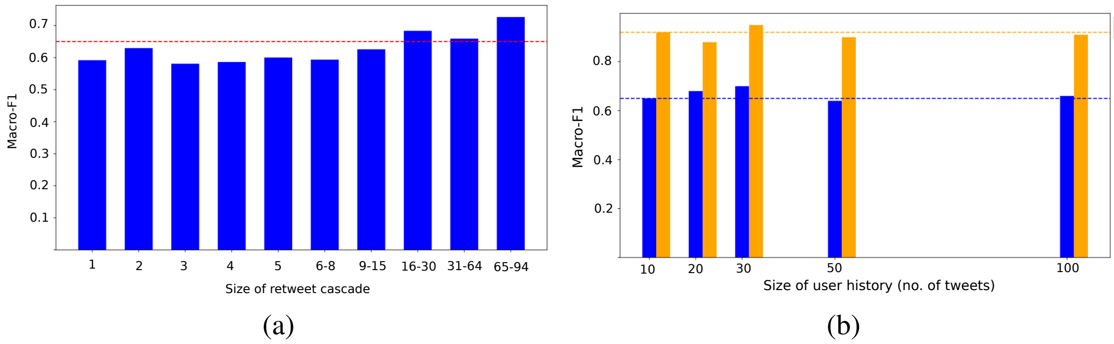


Figure 3.8: (a) Impact of cascade size in performance of RETINA-S (blue). (b) Impact of historical activity of users on the performance of RETINA-S (blue) and RETINA-D (yellow). The dashed lines signify the overall macro-F1.

establishes that RETINA-S performs better with increasing size of the cascade, i.e., more access to diffusion information.

Impact of history activity. In addition, we also vary the number of historical tweets influencing user behaviour. Figure 3.8(b) highlights that the performance of RETINA in both static and dynamic settings first increases by varying history size from 10 to 30 tweets and then drops or remains the same.

3.2.8 Limitations and future work

- **Partial diffusion network.** Owing to API rate limits and our computing resources, we could obtain the follower network only to a depth of 3-hops. As such, the topological information contains missing links, which impacts the performance of information diffusion models that heavily rely on network features.
- **Beyond organic diffusion.** At times, the content users engage with can be part of the 'recommended stream'. Incorporating such information without explicit markers is intractable. Research into how the diffusion of 'paid promotions' varies from organic ones and how the collusive 'paid' setups (Dutta and Chakraborty, 2022) impact the spread of hate are interesting directions for future work.

- **Beyond featured-engineered detection of hate.** While motivated by observations from the data, the current system includes highly engineered features, especially for hate speech detection. Given that more updated PLM-based setups are now available, we need to develop a mechanism to incorporate them. However, our research and existing literature show that PLMs are not good at incorporating non-textual, i.e., numerical signals. Hence, effective modelling of network-level information with PLMs still needs to be explored.

3.3 Probing dynamics of PLMs for hate detection

3.3.1 Motivation

Research gap. Around the same time, we employed a feature-engineered approach to analysing hate speech, and PLMs gained prominence. With the arrival of the transformer architecture (Vaswani *et al.*, 2017), hate speech tasks also gained a significant boost (Mathew *et al.*, 2021; Caselli *et al.*, 2021*a*; Masud *et al.*, 2022). However, the choice of the PLM employed for hate detection is often arbitrary and relies on default hyperparameters (Sun *et al.*, 2019*a*). Despite PLMs being prone to variability in performance (Sellam *et al.*, 2022), there is limited research comparing the pertaining and finetuning settings for subjective tasks like hate speech detection. While previous studies on hate speech have performed hyperparameter tuning, they examine either a single architecture (Founta *et al.*, 2019), a single PLM (Vidgen *et al.*, 2021*b*), or a single dataset (Mathew *et al.*, 2021). A holistic examination of how different PLMs and datasets in hate speech interact during finetuning is missing from the literature.

Research questions. Figure 3.9 provides an overview of our research questions (RQ). We broadly study two critical aspects of PLMs by analysing the impact of different pretraining and finetuning strategies over the following RQs:

RQ0: How do variations in weight initialisation impact hate detection?
RQ1: How do variations in saved checkpoints impact hate detection?
RQ2: Does newer pretraining data impact downstream hate detection?
RQ3: Does the classifier head complexity impact hate detection?
RQ4: What impact do individual/grouped layers have on hate detection?

Note. Due to BERT and RoBERTa intermediate checkpoints (Sellam *et al.*, 2022; Elazar *et al.*, 2023) employed in RQ1-RQ2 being available only in English, we are constrained to datasets only in English. While non-English datasets can be utilised to some extent in RQ3 and RQ4, we are again constrained due to BERTweet (Nguyen *et al.*, 2020) and HateBERT (Caselli *et al.*, 2021*a*) variants being available only in English.

Contribution summary. To the best of our knowledge, we are the first to evaluate PLMs' learning dynamics for hate speech detection (Masud *et al.*, 2024*b*). One of our work's core contributions is to examine five different PLMs and seven English datasets under one study. Consequently, we observe that the dynamics of PLMs for hate detection differ significantly from the other use cases (Sellam *et al.*, 2022; Durrani *et al.*, 2022). There are exciting trends in pretraining learning dynamics, with peaks at early checkpoints. On the finetuning end, we observe that, unlike BERT, the last layer is not

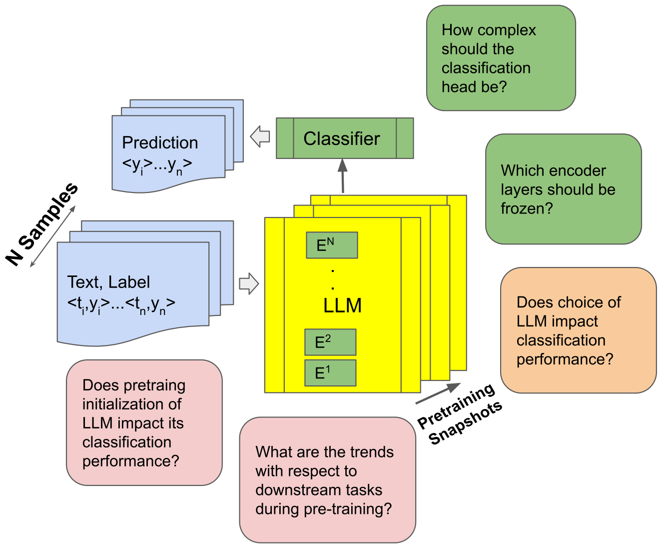


Figure 3.9: A typical PLM-inspired pipeline involves working with one or more checkpoints, i.e., PLM model weights obtained after pretraining. The checkpoint is then finetuned for downstream tasks by keeping one or more layers of PLM trainable along with a trainable classification head (CH). Finally, the PLM + CH generates predictions on incoming test samples.

Table 3.4: Datasets employed in this study. Abbreviation: H: Hate, NH: Not Hate, OFF: Offensive, NOT: Not Offensive. Datasets with * have a predefined train-dev-test split. For others, we take a 75-25% split for train-test sets, with another 25% of the train reserved as a development set.

Dataset	Platform of origin	Time of collection	Train	Dev	Test
`Waseem` (Waseem and Hovy, 2016)	Twitter	Prior to Jun '16	6077	2026	2701
`Davidson` (Davidson *et al.*, 2017)	Twitter	Prior to Mar '17	13940	4647	6196
`Founta` (Founta *et al.*, 2018)	Twitter	March '17 - April '17	33293	11098	14798
`OLID`* (Zampieri *et al.*, 2019*b*)	Twitter	Prior to Jun '19	9930	3310	860
`HateXplain` (Mathew *et al.*, 2021)	Twitter & Gab	Jan '19 - June '20	11303	3768	5024
`Dynabench` (Vidgen *et al.*, 2021*b*)	Synthetic (human-generated)	Sept '20 - Jan '21	23143	7715	10286
`Toxigen` (Hartvigsen *et al.*, 2022)	Synthetic (LLM-generated)	Prior to Jul '22	141159	47054	62738

the most effective for hate detection via mBERT. Our analysis reveals the limited benefit of employing a more recent pretraining corpus. We further call into question the use of domain-specific models and highlight the need for dynamic datasets for benchmarking hate speech.

3.3.2 Experimental setup

Datasets for evaluation. We utilise seven publicly available hate detection datasets in English (Table 3.4). `Waseem`, `Founta`, `Davidson` and `OLID` are chosen based on their prominence in literature. More recent datasets, such as `HateXplain` as well as synthetically generated ones (either by humans, like `Dynabench` or by LLMs, like `Toxigen`), are also picked. As `HateXplain` has multiple annotator responses for each sample, we consider those samples as hateful, where a majority of annotators label them as either hate or offensive.

Note on dataset characteristics. Data drift (Lu *et al.*, 2019) measures the change in feature space between two dataset versions. All samples in the old (*aka* source) dataset

Table 3.5: Data drift measuring the lexical difference in macro-F1 %.

Dataset	Davidson	Dynabench	Founta	HateXplain	OLID	Toxigen	Waseem
Davidson	0.00						
Dynabench	62.60	0.00					
Founta	70.26	59.47	0.00				
HateXplain	66.23	64.12	71.91	0.00			
OLID	63.66	74.21	80.82	80.82	0.00		
Toxigen	91.09	85.88	80.86	91.70	94.76	0.00	
Waseem	69.47	79.06	84.59	67.70	57.20	96.00	0.00

Table 3.6: Overview of PLMs employed in this study. YoR is the year of release (either the public model or the source research paper). We also enlist the data source employed for training. The systems use masked language modelling (MLM) and next-sentence prediction (NSP) as pretraining strategies.

Model	YoR	Pretraining Dataset	Pretraining strategy
BERT (Devlin *et al.*, 2019)	2018	Book Corpus & English Wikipedia	MLM + NSP
mBERT (Devlin *et al.*, 2019)	2018	BERT Pretrained on all Wikipedia data for 104 languages with the most representation in Wikipedia	MLM + NSP
HateBERT (Caselli *et al.*, 2021*a*)	2020	RAL-E (Reddit Comments) - 1.5M Comments	Retrained BERT with MLM Objective
BERTweet (Nguyen *et al.*, 2020)	2020	850M Tweets	Only MLM
RoBERTa (Liu *et al.*, 2019)	2019	Book Corpus, Common Crawler, WebText & Stories	Dynamic MLM + NSP

are labelled as 0 for analysis. Consequently, all samples in the new (aka target) dataset are labelled as 1. A simple classifier is trained to predict the labels as {0,1}. A high performance indicates discriminatory features between the two versions of the dataset. Following the concept of data drift, we assess how distinguishable the above datasets are from each other. A simple frozen BERT-based classifier is employed as the base model. From Table 3.5, we observe that, on average, the datasets are differentiable on the latent space with a macro-F1 of 60-80%. `Toxigen` being machine-generated was more distinguishable than the rest, with a macro-F1 of 85-90%, yet in our later observations, it does not show significant deviations in patterns for the RQs.

PLMs for evaluation. We focus on the open-sourced BERT (Devlin *et al.*, 2019) and RoBERTa (Liu *et al.*, 2019) family of models (PLMs), which are actively employed in hate speech detection (Antypas and Camacho-Collados, 2023) in English. Here we test MultiBERT (Sellam *et al.*, 2022), OLM-RoBERTa and mBERT (Devlin *et al.*, 2019). We also compare domain-specific PLMs like BERTweet (Nguyen *et al.*, 2020) and HateBERT (Caselli *et al.*, 2021*a*) against BERT. An overview of the various PLMs (*aka* backbone models) employed in this study is outlined in Table 3.6.

Classification head (CH). To account for CHs of varying complexity, we introduce:

- *Simple CH*: A linear layer followed by Softmax.
- *Medium CH*: Two linear layers with intermediate $dim = 128$ and intermediate activation function as ReLU, followed by a Softmax.
- *Complex CH*: Two linear layers with an intermediate $dim = 512$, ReLU activation, and an intermediate dropout layer with a dropout probability of 0.1, followed by a softmax layer. We borrow this setup from Ilan and Vilenchik (2022).

Finetuning setup. In order to reduce the variability that encoder layers can induce in the final output, we freeze the encoder layers and only finetune CHs.

Random seeds. We use three random seeds hereby referred to as the *MLP seeds*

($ms = \{12, 127, 451\}$) to initialize the CHs. All our RQs report individual as well as aggregated results over ms.

Text preprocessing. We remove emojis, punctuations, and extra whitespaces to preprocess the textual content. URLs and usernames (beginning with '@') are also replaced with <URL> and <USER>, respectively.

Hyperparameters. All experiments are run with NVIDIA RTX A6000 (48GB), RTX A5000 (25GB) & Tesla V100 (32GB) GPUs. Significance tests are run with a random seed value of 150. We employ the two-sided t-test and Cohen-d to measure the effect size. The setup employs PLMs that are publicly available on HuggingFace. The classifiers use AdamW optimiser with a batch size of 16 and sentences padded to a max length of the respective PLM. We keep the learning rate (LR) at 0.001 (for all RQs) to be in line with the default Adam-W optimiser setting in Huggingface's implementation. We use a linear scheduler with a warmup for the optimiser. The classifiers are trained for two epochs for all our experiments.

Evaluation metrics. Owing to class imbalance we report macro-F1 for all settings.

Note on RQ0. The role of random weight initialisation for PLM pretraining and finetuning has been adequately studied in the literature (Sellam *et al.*, 2022). Our preliminary analysis also corroborates that random seed initialisations lead to variation in performance. The results of this analysis are available in Appendix B.1.

3.3.3 Impact of variations in saved checkpoint

Hypothesis. To study the impact of intermediate checkpoints on downstream tasks, Elazar *et al.* (2023) released 84 intermediate pretrained checkpoints, one for each training epoch of RoBERTa. Employing these intermediate checkpoints, we raise the question, "how do variations in saved checkpoints impact hate detection?" This question is necessary as we hypothesise the model's performance will grow during the early checkpoints and then saturate. It should allow one to find a sweet spot to pretrain task-specific PLMs for a shorter duration, saving compute resources.

Setup. Provided by Elazar *et al.* (2023), we employ the 84 RoBERTa pretrained checkpoints ($C_n \in C_1, C_2, \ldots, C_{84}$). Each pretrained checkpoint PLM is frozen (set to untrainable), and simple and complex CHs are trained separately for each ms.

Table 3.7: The nth checkpoint (C_n) which leads to maximum macro-F1 for simple and CHs recorded over $S_i \in ms$.

Dataset	Simple			Complex		
	S_{12}	S_{127}	S_{451}	S_{12}	S_{127}	S_{451}
Waseem	C_3: 0.660	C_3: 0.668	C_2: 0.691	C_2: 0.734	C_2: 0.738	C_2: 0.756
Davidson	C_2: 0.739	C_2: 0.740	C_2: 0.775	C_2: 0.824	C_3: 0.810	C_2: 0.764
Founta	C_3: 0.870	C_2: 0.861	C_3: 0.869	C_2: 0.879	C_2: 0.880	C_2: 0.878
OLID	C_2: 0.660	C_2: 0.649	C_2: 0.654	C_2: 0.667	C_2: 0.693	C_2: 0.672
HateXplain	C_2: 0.646	C_2: 0.666	C_4: 0.647	C_2: 0.694	C_2: 0.672	C_2: 0.700
Dynabench	C_2: 0.626	C_2: 0.629	C_2: 0.625	C_2: 0.627	C_2: 0.623	C_2: 0.631
Toxigen	C_2: 0.733	C_2: 0.732	C_2: 0.733	C_2: 0.764	C_2: 0.763	C_2: 0.764

Findings. The general trend indicates that each checkpoint possesses hate detection capacity to varying degrees; contrary to our hypothesis, we observe the performance for hate speech detection peaks early (mostly around checkpoint 2) and then rapidly declines. This trend is consistent across different datasets, seeds, and CH complexity as

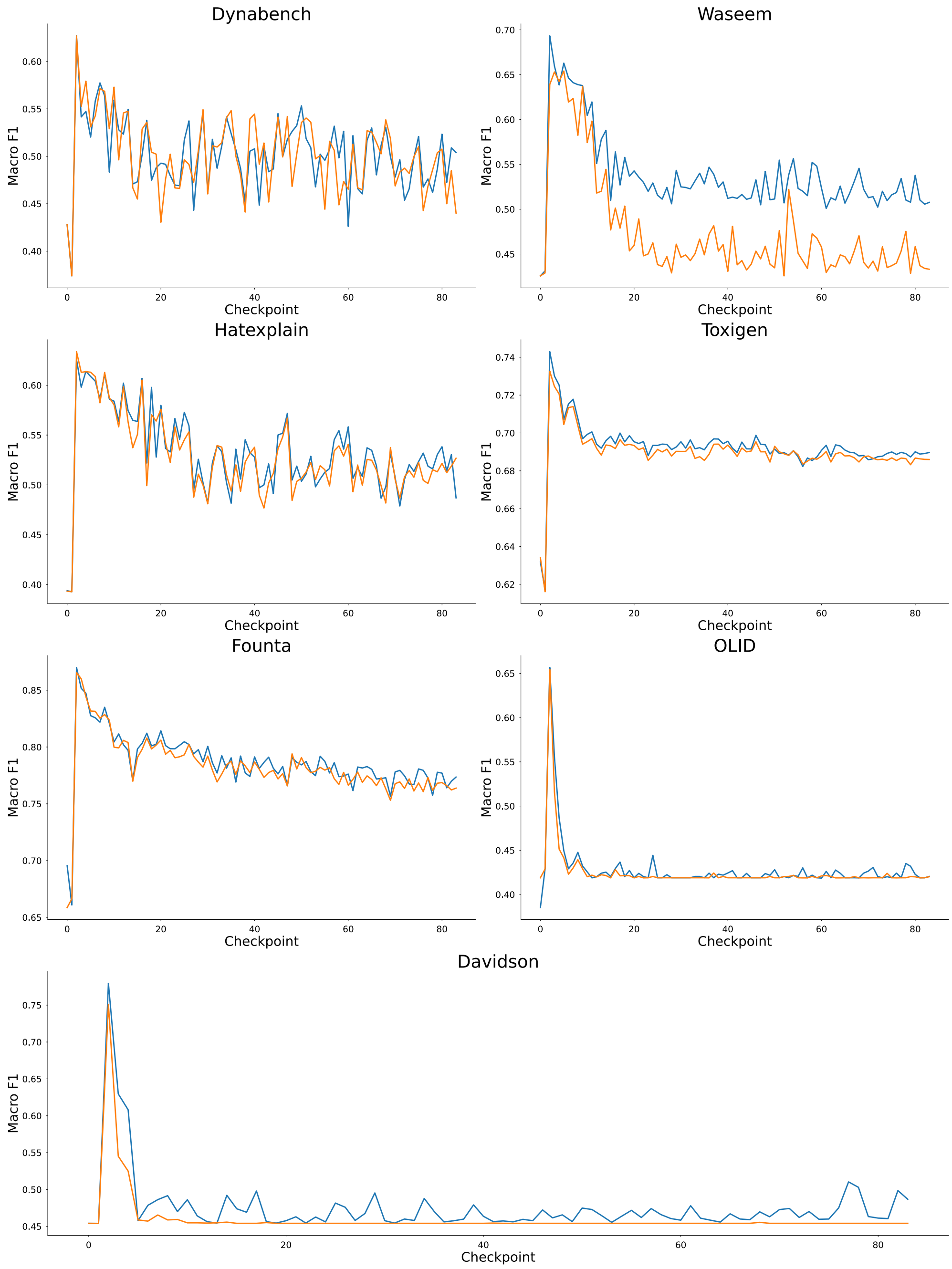


Figure 3.10: Macro-F1 (averaged over ms) attained when finetuning is done on the $n^{th} \in 1, \cdots, 84$ checkpoint (C_n). We report for simple (yellow) and complex (blue) CHs. Performance peaks with early checkpoints around C_n are clearly visible for all configurations.

captured by the highest macro-F1 reported in Table 3.7 and Figure 3.10. While simple and complex CHs follow the same overall pattern, a significant difference in maximum macro-F1 is obtained at each checkpoint between the two setups. The same is recorded in Table 3.8. We observe that for 5 datasets, there is a significant improvement in macro-F1 score when employing complex CH instead of simple. We corroborate our analysis of the superiority of early checkpoints with varying learning rates (LR), – 0.001 (default), 0.01, and 0.1. Averaged across ms, we observe that for a given quadruple <dataset, learning rate, checkpoint, CH>, checkpoint #2 is consistently at par with checkpoint #3, as highlighted by the difference (diff) row in Table 3.9.

Table 3.8: Comparison of maximum macro-F1 obtained under varying ms for the simple (Sim.) and complex (Com.) CHs. ES stands for effect size. ** and * indicates whether the difference in maximum macro-F1 is significant by ≤ 0.05 and ≤ 0.001 p-value, respectively.

Dataset	**12**			**127**			**451**		
	Sim. F1	**Com. F1**	**ES**	**Sim. Max F1**	**Com. F1**	**ES**	**Sim. Max F1**	**Com. F1**	**ES**
`Waseem`	C_3: 0.660	C_2: 0.734	0.581**	C_3:0.668	C_2:0.738	0.547**	C_2: 0.691	C_2:0.775	0.580**
`Davidson`	C_2: 0.739	C_2: 0.824	0.953**	C_2:0.740	C_3:0.810	0.852**	C_2: 0.775	C_2:0.764	0.113
`Founta`	C_3: 0.871	C_2: 0.879	0.278*	C_2:0.861	C_2:0.880	0.613**	C_3: 0.869	C_2:0.878	0.269
`OLID`	C_2: 0.661	C_2: 0.667	0.110	C_2:0.649	C_2:0.694	0.242	C_2: 0.654	C_2:0.672	0.164
`HateXplain`	C_2: 0.640	C_2: 0.687	0.599**	C_2:0.659	C_2:0.665	0.088	C_4: 0.640	C_2:0.694	0.751**
`Dynabench`	C_2: 0.626	C_2: 0.628	0.010	C_2:0.629	C_2:0.623	0.123	C_2: 0.625	C_2:0.631	0.118
`Toxigen`	C_2: 0.733	C_2: 0.764	1.810**	C_2:0.732	C_2:0.763	1.772**	C_2: 0.733	C_2:0.764	1.835**

Table 3.9: Macro-F1 for C_2 and C_3 with varying LRs and CHs. Diff ($C_2 - C3_3$) depicts the difference in the performance of two checkpoints.

CH	**Checkpoints**	**LR**	**`Davidson`**	**`Dynabench`**	**`Founta`**	**`HateXplain`**	**`OLID`**	**`Toxigen`**	**`Waseem`**
Simple	C2	0.001	0.75	0.63	0.867	0.657	0.653	0.73	0.637
	C3	0.001	0.547	0.553	0.86	0.62	0.517	0.72	0.653
	Diff (C2-C3)		0.203	0.077	0.007	0.037	0.136	0.01	-0.016
Complex	C2	0.001	0.78	0.627	0.88	0.687	0.677	0.76	0.743
	C3	0.001	0.763	0.577	0.857	0.613	0.55	0.74	0.69
	Diff (C2-C3)		0.017	0.05	0.023	0.074	0.127	0.02	0.053
Simple	C2	0.01	0.813	0.493	0.827	0.683	0.657	0.73	0.743
	C3	0.01	0.76	0.52	0.843	0.543	0.623	0.72	0.72
	Diff (C2-C3)		0.053	-0.027	-0.016	0.14	0.034	0.01	0.023
Complex	C2	0.01	0.837	0.593	0.863	0.623	0.617	0.73	0.753
	C3	0.01	0.643	0.517	0.867	0.617	0.597	0.72	0.723
	Diff (C2-C3)		0.194	0.076	-0.004	0.006	0.02	0.01	0.03
Simple	C2	0.1	0.75	0.52	0.777	0.62	0.577	0.72	0.75
	C3	0.1	0.76	0.543	0.823	0.517	0.567	0.717	0.68
	Diff (C2-C3)		-0.01	-0.023	-0.046	0.103	0.01	0.003	0.07
Complex	C2	0.1	0.76	0.35	0.487	0.543	0.527	0.447	0.677
	C3	0.1	0.45	0.35	0.57	0.467	0.42	0.433	0.71
	Diff (C2-C3)		0.31	0	-0.083	0.076	0.107	0.014	-0.033

Takeaways. A fully pretrained model (i.e., 84th checkpoint) might not be necessary for hate speech-related tasks. We concur that this may be due to a mismatch between the model's training on well-written datasets such as Wikipedia and Book Corpus and the noisy nature of hate speech curated from OSNs. *When the model has not yet fully learned the English language syntax, it could be better suited to capture the noisy information in the hate speech text.* However, a more extensive examination of broader hate speech-related tasks and languages will be required to ascertain this. Our first recommendation is to make intermediate checkpoints available if pretraining from scratch or continued is involved. If we assume the same training setup as used by (Elazar *et al.*, 2023) and if the pretraining is stopped after 8-10 epochs, noticing the performance drop on the downstream tasks, then 8-10× compute could be saved. The direction then will be to formulate intermittent GLUE benchmarks for evaluation at intermediate epochs.

Does early performance peak for hate speech classification also indicate that the notion of "hatefulness" is adequately captured early? The findings in RQ1 necessitate the examination of parametric knowledge vs performance.

3.3.4 Impact of newer pretraining data

Hypothesis. Given the dynamic nature of hate speech, we hypothesise that PLMs pretrained on more recent world knowledge should be better hate speech detectors.

Setup. For this analysis, we use chronologically updated checkpoints of RoBERTa released by Online Language Modelling[2] (OLM). Initiated by HuggingFace, OLM is a repository of updated PLM models and tokenisers that are pretrained on regular and latest world knowledge snapshots obtained via Common Crawl and Wikipedia. The initiative aims to induce explicit knowledge of newer concepts and updated factual information in the PLMs. At the time of compiling this research, the OLM repository had snapshots of 6 LMs and 19 datasets. Out of these, only two RoBERTa models were in succession in October 2022 (R_{O22}) and December 2022 (R_{D22}). We compare these variants against RoBERTa initially released in June 2019 (R_{J19}).

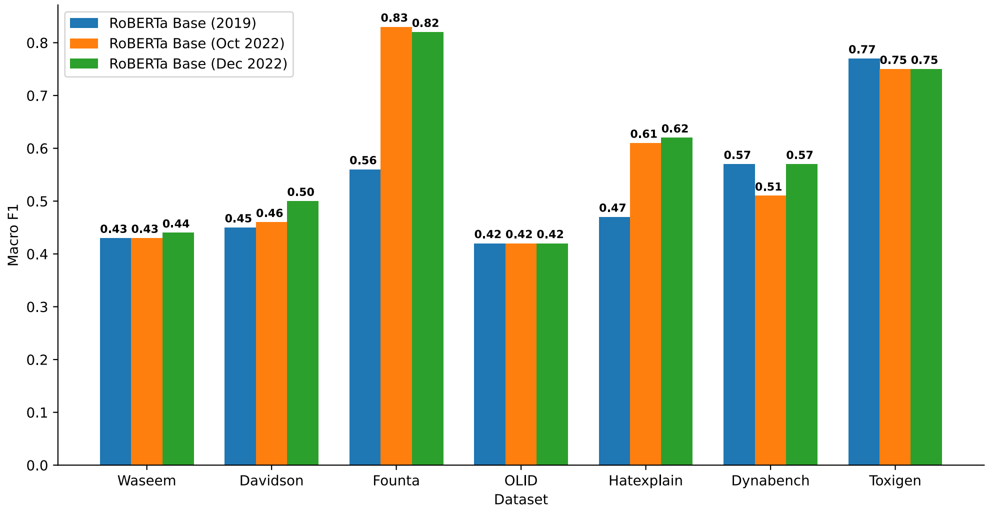


Figure 3.11: Macro-F1 on different datasets finetuned with an MLP classifier on RoBERTa variants. The variants employed are from June 2019 (R_{J19}), October 2022 (R_{O22}), and December 2022 (R_{D22}). Each variant is trained on a training corpus from Wikipedia, and Common-Crawl is curated and updated before the date associated with the model. R_{J19} is the original RoBERTa model and R_{O22} and R_{D22} are its more recent variants.

Findings. The performance comparison in Figure 3.11 reveals that half of the datasets (4/7) do not showcase a significant change in performance with newer PLMs. It should be noted that in order to assess the impact of differently updated PLMs on downstream hate detection, the performance should be interpreted at the individual dataset level and not across datasets. The findings can be attributed to the fact that most of the datasets were curated years ago (Table 3.4) and are on the explicit side of hate. Consequently, events present in these datasets were already sufficiently represented in the pretraining corpus of the original model (R_{J19}). Only three datasets register a sharp jump in performance, and a 25 macro-F1 jump for `Founta` is suspicious. Previous literature (Zhu *et al.*, 2023) has postulated that a substantial improvement in NLP performance, such as in `Founta` can be an indicator of data leakage. On the other hand, it is interesting to note that the two synthetically generated datasets, `Dynabench` and `Toxigen`, do not record any significant deviation from overall trends, even though `Dynabench` is human-generated while `Toxigen` is machine-generated.

Takeaways. *The findings underscore the need to develop more dynamic and updated hate speech datasets.* As data curation is labour-intensive, using computation techniques to generate synthetic datasets and pseudo-labels is a direction that needs further

[2] https://huggingface.co/olm

exploration in the toxicity literature (Bhutani *et al.*, 2024; Hartvigsen *et al.*, 2022).

3.3.5 Impact of the complexity of the classifier head

Hypothesis. When finetuning, most downstream tasks employ a simple CH to retain maximum latent information from the PLMs. It calls for an examination of the relationship between PLMs and CHs with respect to hate speech detection. We hypothesise that employing a relatively complex CH should perform better than its simpler counterpart (Ilan and Vilenchik, 2022).

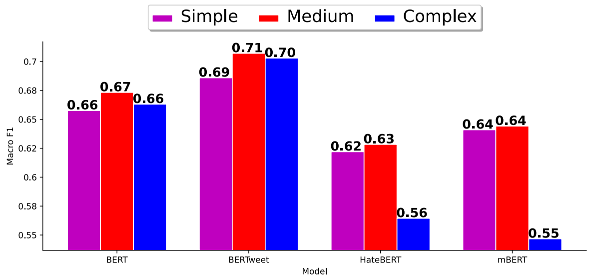
Setup. We run our experiments on three CHs of increasing complexity – simple, medium, and complex (described in Section 3.3.2). The pretrained model is frozen for this set of experiments to capture the variability introduced by the trainable CH's complexity. For a fair comparison, experiments are run on four BERT variants – BERT, BERTweet, HateBERT, and mBERT.

Findings. Figure 3.12 provides an overview of the impact of CH architecture on the finetuning performance. Granular results controlling for ms are enlisted in Appendix B.2. Overall, compared to a simple CH, a more sophisticated one (either medium or complex) is better. Among different variants, one observes a slight decrease in performance across datasets using mBERT. Surprisingly, BERTweet, a relatively lesser-used PLM for hate speech detection, outperforms its supposedly superior domain-specific counterpart, HateBERT. While HateBERT is continued pretrained on offence subreddits, BERTweet is continued pretrained on tweets. Given that most hate speech datasets are either directly drawn from Twitter or synthesised in a short-text fashion, BERTweet could be indirectly capturing both short-text syntax and offence from Twitter. Finally, comparing HateBERT with BERT, one observes that HateBERT's performance is highly dependent on the CH, with a more complex one often needed to bring it to par with its coevals. Meanwhile, BERT with a complex CH demonstrates comparable performance to domain-specific PLMs (both HateBERT and BERTweet) and even outperforms them in several cases. *Interestingly, we observe that a general-purpose pretrained PLM with a complex CH can mimic the results of a domain-specific one.*

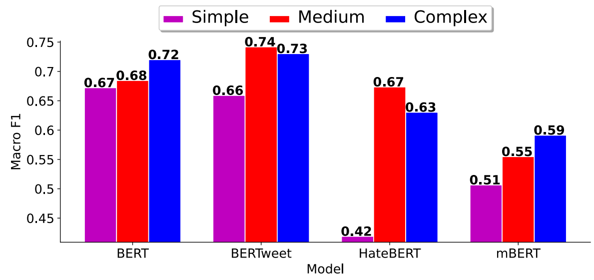
Takeaways. Given the variability in hate speech detection introduced by CHs, even when the underlying PLM is frozen, calls for more research in establishing the role of CH complexity in the finetuning of NLP tasks. If the results are similar for a broad range of tasks, it questions the resource allocation for curating domain-specific PLMs. When even a random set of test samples can help steal model weights (Krishna *et al.*, 2020) in NLP tasks, it points to limited domain-specific learning in light of the adversary. Thus, more experiments are needed to establish their superiority over general-purpose models.

3.3.6 Impact of individual/grouped layers

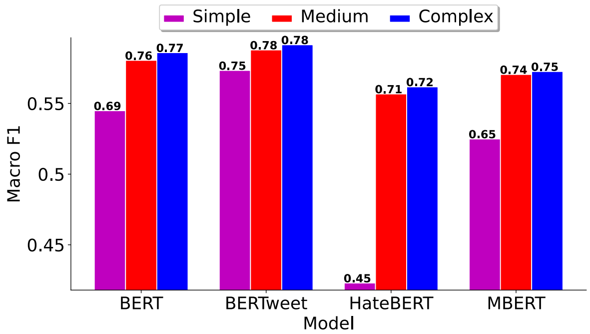
Hypothesis. Our prior RQs froze all the encoder layers. Meanwhile, different layers or groups of layers in a PLM impact downstream performance to a varying degree (de Vries *et al.*, 2020; van Aken *et al.*, 2019). During finetuning, the PLM layers closer to the CH capture the maximum task-specific information (Durrani *et al.*, 2022). Specifically, setting the lower layers' parameters untrainable is a common (Durrani *et al.*, 2022) finetuning practice.



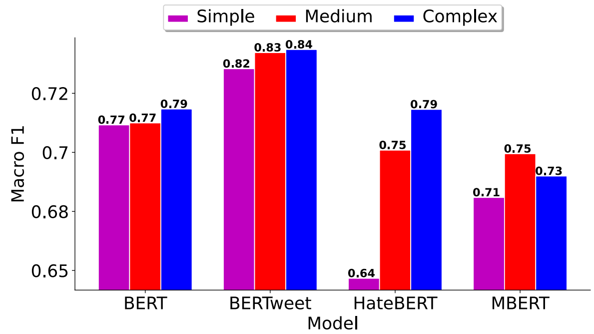
(a) Dynabench



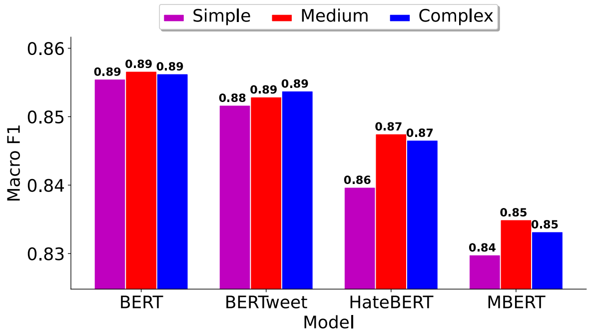
(b) OLID



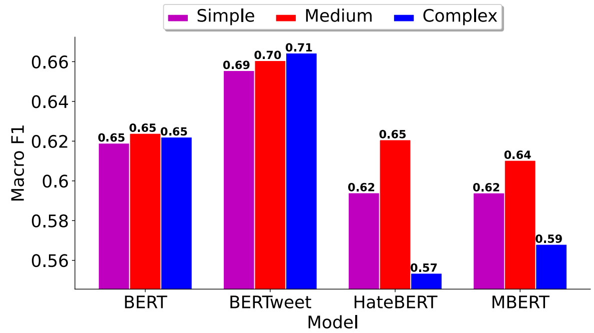
(c) Waseem



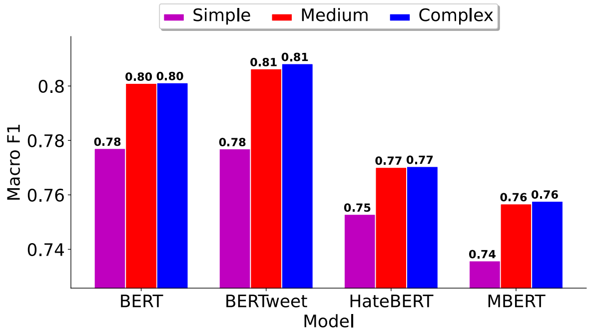
(d) Davidson



(e) Founta



(f) HateXplain



(g) Toxigen

Figure 3.12: Macro-F1 scores (averaged over ms) examining BERT-variants (BERT, BERTweet, HateBERT, and mBERT) with CHs of varying complexity.

While layer-wise analyses have been explored in various NLP tasks (de Vries *et al.*, 2020; van Aken *et al.*, 2019), a comprehensive examination across models, datasets, and finetuning scenarios has been notably absent in the hate speech domain. We hypothesise that different layers or groups of layers in the PLM will be of varying importance for hate detection. Borrowing from the popular finetuning settings documented by Sun *et al.* (2019*a*), one expects training the last few (higher) layers to yield better performance than training earlier (lower) layers. Further, the setting where more layers are trainable is likely better, giving the model more ability to learn the latent space.

Setup. We freeze all parameters except the probed layer, and the CH is initialised with ms. We probe the impact of layers beginning with the analysis of setting (un)trainable individual layers $L_1, L_2, \ldots, L_{12}$ and then setting (un)trainable groups of layers, *aka* region. A 12 layer PLM comprises 4 regions (R_1, R_2, R_3, R_4) of 3 consecutive layers with $R_1 = \{L_1, L_2, L_3\}$ and so on. For the layer-wise case, the CH is placed on top of the trainable layer. Similar to RQ3 (Section 3.3.5), experiments in this setup are also run on four BERT variants – BERT, BERTweet, HateBERT, and mBERT

Findings. Figure 3.13 shows that trainable higher layers (closer to the CH) lead to higher macro-F1 for most BERT-variants. However, no single layer emerges as a clear winner across all datasets and models. Overall, the trend for higher layers leading to substantially better performance holds significantly for 5 out of 7 datasets and partially for `Founta`. Interestingly, the notion of higher layers being important does not hold for mBERT. For mBERT, layer #4 seems to dominate across datasets. While obtaining the best performance from the middle layers of PLMs is counterintuitive, similar behaviour regarding mBERT has been reported in other NLP tasks (de Vries *et al.*, 2020; Müller-Eberstein *et al.*, 2022). We postulate this behaviour stems from mBERT's need to be simultaneously equally generalised vs informative for all languages.

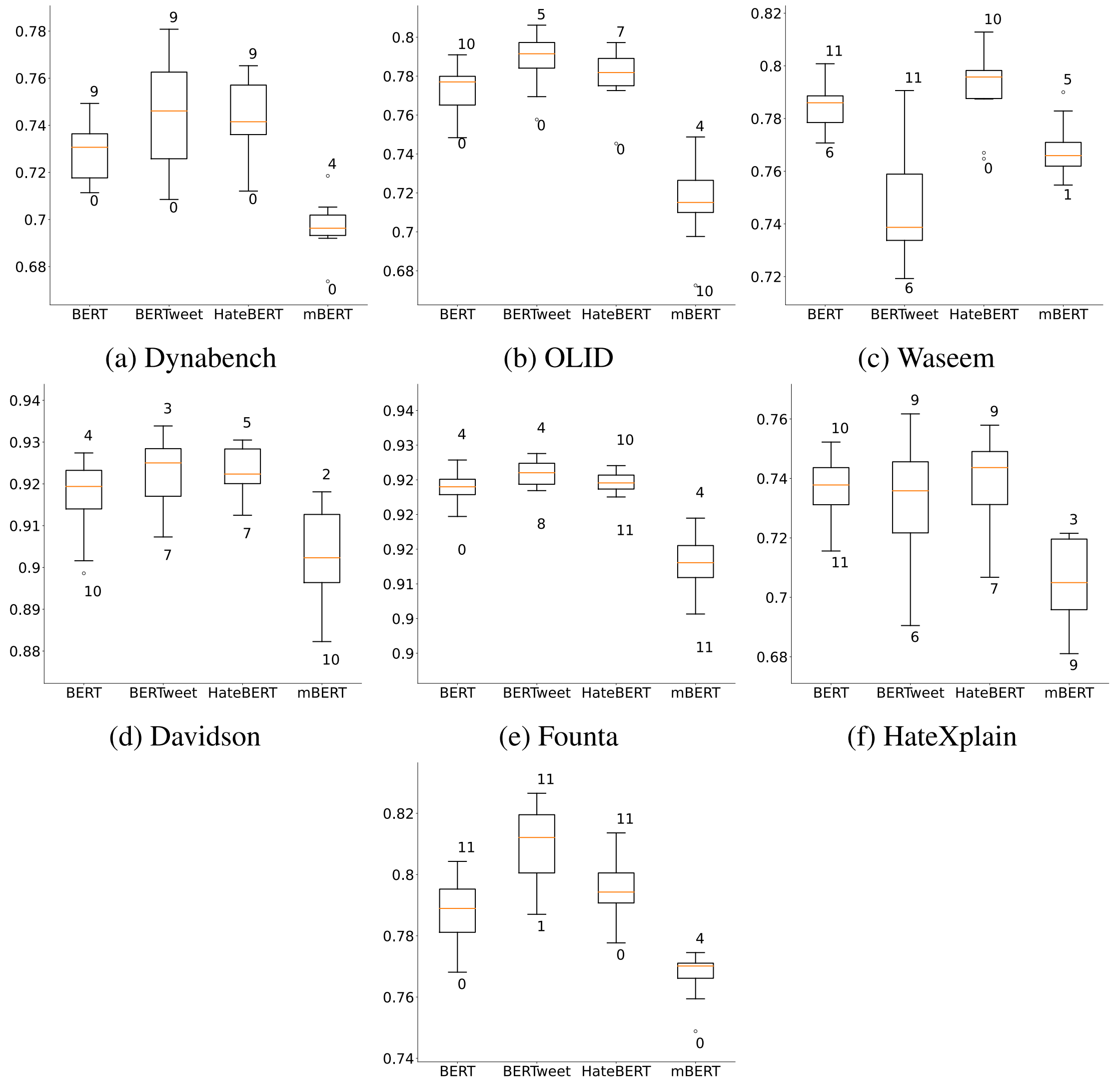
As layer-wise ablations are computationally expensive, we perform region-wise analysis to assess if they can provide equivalent coarse-grained information. Again, analogous to layer-wise analysis, no region dominates significantly across all datasets as Figure 3.14 records. Similar to layer-wise results, our findings on region-wise analysis indicate that training the last region performs better than the other settings, where only a specific region is trained. Surprisingly, training only R3 also results in considerable performance gains. Consequently, when the last region is frozen, it is never the best combination for any dataset or model, further validating the status quo.

Table 3.10: Macro-F1 based on BERTweet cross-dataset generalisation. In each row, two columns with the same dataset name as the one in the row corresponding to the in-domain evaluation. The others correspond to out-of-domain evaluation.

Test	Train			
	OLID Min	**OLID Max**	**Dynabench Min**	**Dynabench Max**
OLID	0.747	0.817	0.435	0.520
Dynabench	0.435	0.491	0.705	0.783

Based on the best seed, layer, and PLM combinations obtained in Figure 3.13, we randomly pick `Dynabench` and `OLID` to perform a cross-dataset generalisation experiment and examine the impact of hyperparameters associated with minimum and maximum in-domain PLM. Coincidentally, it is BERTweet in both cases. In line with previous studies on cross-dataset generalisation in hate speech (Fortuna *et al.*, 2021), we observe a poor performance on out-of-distribution testing. The silver lining from Table 3.10 is that our results do hint that the best finetuning setting also corresponds to the best out-of-domain generalisation. *Such settings can be useful to narrow down the hyperparameter search in balancing in-dataset vs. out-of-dataset performance gains.*

Takeaways. Appendix B.3 provides the seed-wise results for layer and region (un)freezing experiments. Overall, while our experiments with layers and regions reinstated the status quo of finetuning the last few layers to obtain the best performance,



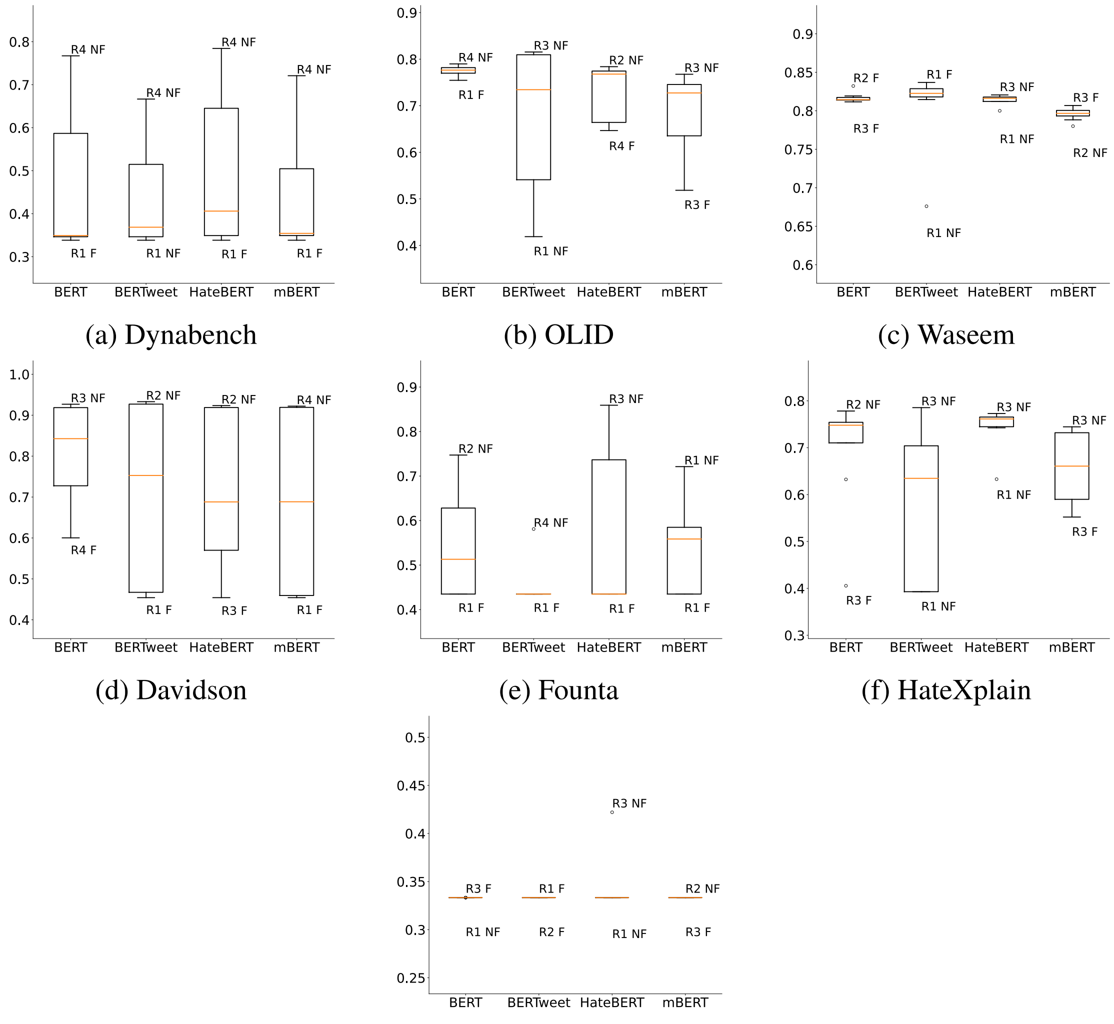
(a) Dynabench (b) OLID (c) Waseem

(d) Davidson (e) Founta (f) HateXplain

(g) Toxigen

Figure 3.13: Descriptive statistics of macro-F1 when training only the $L_i$the layer during finetuning. The results are averaged across the ms.

it only holds for monolingual and domain-specific models. As the middle layers appear more critical in mBERT, NLP tasks, including hate speech detection that employs mBERT, should start with keeping the middle layers unfrozen for finetuning. The region-wise performance of PLMs appears to be a characteristic of the underlying PLM and is less impacted by variation in datasets. Such intuitions can help narrow the experiments one has to run to obtain better classification configurations for other NLP tasks. From both RQ3 and RQ4, we observe a slight decrease in performance across datasets comparing mBERT and BERT for English datasets. Given that mBERT has more parameters than BERT (178M vs. 110M in the base version), we suggest not using mBERT unless the hate speech is itself multilingual. It is imperative to note that choosing the language of the test set is easier to control in a closed/offline evaluation environment, and not in deployment. Our findings nudge the question of whether the training resources should be invested in one sub-optimal large-scale multilingual model or multiple smaller-scale, language-optimised models. The former may introduce higher variability in finetuning strategies, as observed for BERT vs mBERT in our case.



(a) Dynabench (b) OLID (c) Waseem

(d) Davidson (e) Founta (f) HateXplain

(g) Toxigen

Figure 3.14: Descriptive statistics of macro-F1 while constraining a region to be frozen (suffix F) or non-frozen (suffix NF) during finetuning. The results are averaged across ms. Region R_1 includes layers L_1 to L_3, R_2 from L_4 to L_6, R_3 from L_7 to L_9 and R_4 from L_{10} to L_{12}.

3.3.7 Limitations and future work

- **Impact of finetuning strategy beyond macro-F1.** Hate speech detection systems are prone to modelling biases that adversarially impact the target communities. Thus, experiments narrowing down best-case finetuning strategies need to extend the analysis beyond macro-F1. It remains an open area of research how multiple objectives (performance vs bias mitigation) interact in the hyperparameter space.
- **Toxicity beyond binary labels.** So far, the thesis has examined hate speech diffusion as well as detection in a binary setup (retweet or not, hate or not); however, hateful behaviour is not black and white; it is instead on a spectrum. On the one hand, we need a better spectrum to capture the degree of hate. On the other hand, we also need to evaluate PLM finetuning strategies under fine-grained or multilabel setups.
- **Expanding to multilingual analysis.** The current research heavily relies on PLMs that are available in English. Revisiting these RQs from a multilingual perspective is another area of open research.

3.4 Multiclass and multilingual hate detection

3.4.1 Motivation

Research gap. In order to increase the coverage of hateful instances, a common vogue is to employ hate lexicons and slur terms. This increase in hate samples comes at the cost of low coverage of nuanced constructs such as provocation or implicitness (ElSherief *et al.*, 2021). While a few neutrally seeded datasets also exist (de Gibert *et al.*, 2018; Basile *et al.*, 2019), they tend to focus more on controversial events (e.g., the Lee Rigby murder) or specific hate targets (e.g., immigrants), which may introduce topic bias and artificially inflate model performance. Besides neutral seeding, there is a lack of multilingual hate speech datasets covering Indian topics (Vidgen and Derczynski, 2021). Although with `ConInHate` we introduce the notion of an Indic setting, the sample posts are still in the English language and carry binary labels. A granular coverage of hatefulness in code-mixed Hinglish is missing. In real-world discourse, hate speech's dynamics, syntax, and semantics are bound to change following new triggers (Gao *et al.*, 2017*a*; Florio *et al.*, 2020), making hate speech detection highly contextual. Thus, identifying and modelling hateful posts calls for going beyond language reliance and better augmentation of PLM finetuning with contextual features.

Research questions. We examine the gap in large-scale neutrally seeded multilingual hate speech datasets. As a first step, we curate `GOTHate`. Through `GOTHate`, we sought to answer the following:

RQ1: How difficult is `GOTHate` to classify given its neutral and multilingual nature?
RQ2: How can contextual signals be effectively incorporated to improve the detection of multilingual hate speech in `GOTHate`?

Contribution summary. To address the research gap in large-scale Hinglish datasets, we curate the `GOTHate`dataset. It contains $\approx 50k$ tweets from $\approx 25k$ unique users encompassing nuanced and realistic aspects of online hateful discourse. `GOTHate` is a conglomerate of tweets from seven assorted topics spanning socio-political events and world affairs. Curated following a neutral seeding schema, it captures linguistic variations in the form of English, Hindi, and Hinglish posts. Each post is annotated with one of the four labels: hate, offensive, provocative, and non-hate. Based on a manual inspection of the anntoated samples, we postulate that `GOTHate` can be challenging for hate speech classification models (Kulkarni *et al.*, 2023). We thus perform a thorough assessment of `GOTHate` and compare it against existing hate speech datasets. Our examination reveals that `GOTHate` is characterised by more cohesion within the topics, resulting in less linguistic, syntactic, and contextual diversity across the labels, better miming real-world online hate speech discourse, and making it hard to classify. To capture the intricacies of `GOTHate`, we propose `HEN-mBERT`. It is a mixture-of-experts variant of mBERT augmented with endogenous signals. To accentuate any hateful bias present in the user's posting history, we integrate their timeline and network interactions. The modelling is further infused with example tweets that help distinguish the latent features between the labels. Empirical results attest that endogenously attended `HEN-mBERT` outperforms mBERT by 5% in hate class macro-F1 (Kulkarni *et al.*, 2023).

Table 3.11: Statistical information of GOTHate with # of tweets (users) per label. The user within a topic or label may not be exclusive to that subset. The labels consist of Hate (H), Offensive (O), Provocative (P), and Neutral (N).

Topic (country of origin)	**Label-wise unique tweets (users)**			
	H	**O**	**P**	**N**
Never Trump Campaign (USA)	5 (5)	273 (222)	536 (334)	1443 (889)
Delhi Roits 2020 (India)	1503 (1142)	2071 (1526)	4776 (3248)	8508 (4795)
Demonetization (India)	574 (495)	915 (750)	1446 (1154)	3643 (2112)
Brexit (UK)	38 (35)	505 (423)	1041 (805)	6543 (3654)
Umar Khalid JNU (India)	70 (63)	2465 (1967)	18 (17)	2922 (2028)
Northeast Delhi Riots 2020 (India)	1182 (1055)	1622 (1441)	1605 (1470)	4364 (3539)
Hindu Lives Matter (USA & India)	343 (216)	219 (167)	1134 (552)	1603 (967)
Overall 51367 (25796)	3715 (2925)	8070 (6161)	10556 (7088)	29026 (16186)

3.4.2 Dataset curation

Primary data. We concentrate on hateful content on X, owing to the ease of extracting publicly available posts and their metadata (back in 2022). Using the Twitter API[3], we revisit the Indic Twitter and compile a corpus of $51,367$ tweets posted by $25,796$ unique users, hereafter referred to as source/root tweets and source/root users, respectively. The data consists of seven socio-political events/topics across 3 geographies (USA, UK, and India) and 3 linguistic variations (English, code-mixed Hinglish, and Devanagari Hindi). Table 3.11 presents the statistics of the annotated GOTHate.

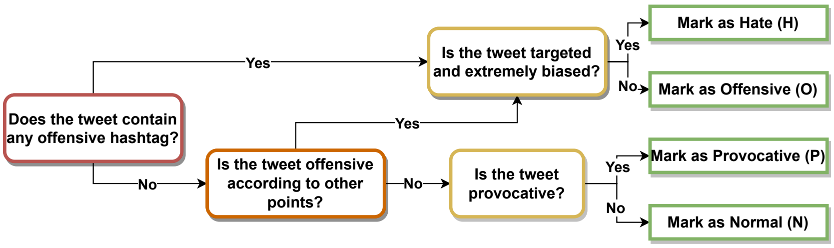
Neutral seeding. Instead of relying on hate lexicons, we collect the tweets using neutral topics, such as *'demonetisation' (India)* and *'Brexit' (UK)*. Our data collection thereby reflects the natural discourse on social media more closely, as hateful and provocative content is interlaced with neutral commentary on a topic. The same user can invoke hateful and non-hateful sentiments depending on the topic under discussion (Masud *et al.*, 2021). Interestingly, we find overlapping posts between the topics of *'Never Trump'* and the Indian protests against the *'Citizenship Amendment Act (CAA)'*. This overlap can be attributed to Trump's visit to India during the demonstrations.

Auxiliary signals. The majority of the posts in the primary dataset are between the years 2019 and 2021. Given the vast range of time, it is not feasible to store news dumps as we did for ConInHate (whose data range was 3 months). Thus, we direct our efforts to look at endogenous signals instead. In order to capture the endogenous influences from X, we also collect the root user's timeline and ego network. The root user's timeline data is curated as the 25 tweets before and after the root tweet was posted. Meanwhile, the ego network consisted of the one-hop followers and followees of the root users, leading to a global network of $\approx 20M$ unique users. To enhance the interaction information, we also collected the latest 100 retweeters of the root tweet, which had retweet cascades. Some of these retweeters are already followers of the root users and are marked as internal retweeters. Retweeters not initially captured in our 1-hop network are marked as external retweeters.

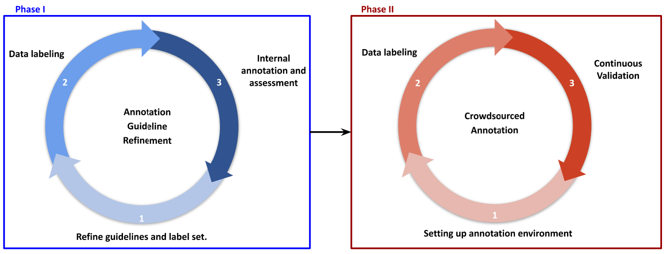
3.4.3 Dataset annotation

We annotate the $50k$ source tweets in a two-phase manner with continuous validation (Founta *et al.*, 2018) as shown in Figure 3.15(b).

[3] https://developer.twitter.com/



(a)



(b)

Figure 3.15: (a) Flowchart for annotating a post. Precedence for labels: Hate (H) > Offensive (O) >Provocative (P) > Neutral (N). (b) Overview of the two-phased continuous-validation annotation process.

Annotation phase I. Two researchers and a lexicographer (referred to as *Group A*) carry out the first annotation phase. They annotate a random set of 1000 root tweets spread across topics. The annotators start with broad definitions borrowing from Davidson *et al.* (2017) and label the posts as hateful, offensive, or neither. During annotation, *Group A* observes that some content had provocative connotations while being neutral as per the Twitter guidelines. We thus introduce 'provocation' as a fourth category. The guidelines improve iteratively until Krippendorf's α (Krippendorff, 2011) of 0.8 is reached.

Annotation phase II. In order to scale the annotations for the entire dataset, we partner with Xsaras[4], a professional data annotation company. Note that before finalising Xsaras, we also ran pilot tests on Amazon Mechanical Turks[5] and Appen[6], out of which Xsaras came across as the closest in terms of topical and linguistic diversity that our dataset supports. After an initial screening, 10 professional annotators (referred to as *Group B*) annotate the rest of the root tweets. The corpus is divided into batches of 2500 samples. The annotation agreement is calculated for 100 random tweets per batch (already annotated by *Group A*) to ensure the annotation quality in each batch. We obtain an average agreement of 0.71 Krippendorf's α.

Annotator details. None of the annotators in either group are associated with any political organisation. All annotators are of Indian origin. They know the socio-political discourse in the USA, UK, and India and are fluent in code-mixed Hindi. All have prior experience using social media and annotating textual datasets. *Group A* consists of three female annotators aged $24 - 34$. *Group B* annotators are within the age 20-64. In *Group B*, 6 annotators are females, and the 4 are males. 6 annotators work on English and Hinglish data, 3 on Hindi (Devanagiri), and 1 on both. The crowdsourced annotators are compensated \$1 per tweet and required to annotate 500 tweets per week.

[4]https://xsaras.com/
[5]https://www.mturk.com/
[6]https://www.appen.com/

Table 3.12: Sample tweets (verbatim) with their annotated labels and reasoning for annotation. These samples are provided to Group B annotators for reference.

Sample	Label	Rationale
Example 1: 'Unko aukaat dikhaenge' (Will show them their place), said a young man with a face covered amidst chanting Jai Shri Ram. Northeast #Delhi saw nothing but a communal riot. People killed, shops and vehicles burnt. #DelhiRiots	H	Refer Hate point 1 (Distorting views of Hindu community)
Example 2: Chincommie virus killing off Chinese. @china leaders killing off Wuhan politicos. Build that wall faster @realDonaldTrump @GOP @CDCgov	H	Refer to Hate Point 1 (demeaning a race. While the Chinese population is not a minority, with regard to this post, they are a vulnerable group. Also note that while Communist China is not hurtful, the term commie like nigger is a derogatory remark)
Example 2: #hindulivesmatter #Hinduphobia Sometimes you just have to jump into activist mode, esp. when #India & #Hinduism are denigrated. An offensive T-shirt has been removed after complaints to the T-shirt company. When the next time such things happen, WILL YOU ACT ?????	P	Refer to Provocative Point 2 (An indirect call to action)
Example 4: #unsungHindus #Hinduunity #hindulivesmatter Sometimes, you just have to jump into activist mode, especially when India and Hinduism are denigrated. Thank you, Mahalakshmi Ganapathy Vijay Kumar Shourie Bannai.	N	This post is similar to the above one and is related to the same event. This post is not provocative because instead of trying to stir action, it is trying to put this in a positive light (based on the available content)

Annotation schema. Table 3.12 lists some examples and the corresponding reasoning for the assigned label. Note that while `GOTHate` is curated based on geography-specific socio-political topics, the tweets obtained for a topic are not geo-tagged. As shown in Figure 3.15(a), to further reduce the disagreement, we introduced an order of priority $H > O > P > N$. *While anyone can offend and provoke anyone, hate only applies if the attack is against a vulnerable group.*

- **Vulnerable group.** *It is defined as people who have historically been oppressed or abused based on religion, caste, country, colour, gender, ethnicity, etc.* For example, Muslims are a vulnerable group in China, whereas in the USA, people of Chinese origin are a minority. Similarly, religion and caste are some known prejudices within the Indian subcontinent. Therefore, based on the content of a post and the vulnerable group it targets, a post can be considered Hateful (H), Offensive (O), Provocative (P), or Normal (N). Note that we do not have a pre-defined list of vulnerable groups, but use the definition to help the annotators be mindful of the task at hand.
- **Hate.** It is differentiated by extreme bias[7] against the target (Waseem and Hovy, 2016) via any of the following:
  1. Negatively stereotypes or distorts views on a vulnerable group with unfounded claims.
  2. Silences or suppresses a member(s) of a vulnerable group.
  3. Promotes violence against vulnerable group member(s).
- **Offensive.** A statement is considered offensive if it conforms to one of the following points:
  1. Uses derogatory words to abuse, curse ("go die," "kill yourself"), sexualise ("f** you," "kiss my a**"), or express inferiority ("useless," "dumb," "crazy") towards an entity, criticising the person and not the action/event.
  2. Compares with demons/criminals/animals either directly or by implication ("Wasn't X bad enough," "Y is p**").
  3. Uses hashtag(s) covering either points #1 or #2. Hashtags like #EUnuch (UK) #NastyNancy (USA), #SellOutHiliary (USA), #PMNautanki (India), #CoronaJihad (India) are hurtful by themselves. Meanwhile, #DelhiViolence,#NeverTrump, #NotMyPM, or #Resign are not offensive just by themselves and need the content of a tweet to determine their label.

[7] UN Definition: `bit.ly/3HoFpjP`

- **Provocative.** If a post itself is not offensive based on the above definitions but invokes a negative reaction from the reader, then it is considered provocative based on any/all of the following:
    1. Accuses a particular group or individual of an event related to issues surrounding the group.
    2. Invokes a call to action to stir a group against the target.
    3. Implies boycotting the target. The boycott can be social (like disallowing entry), economic (like denying financial entitlements), or political (like denying political participation).

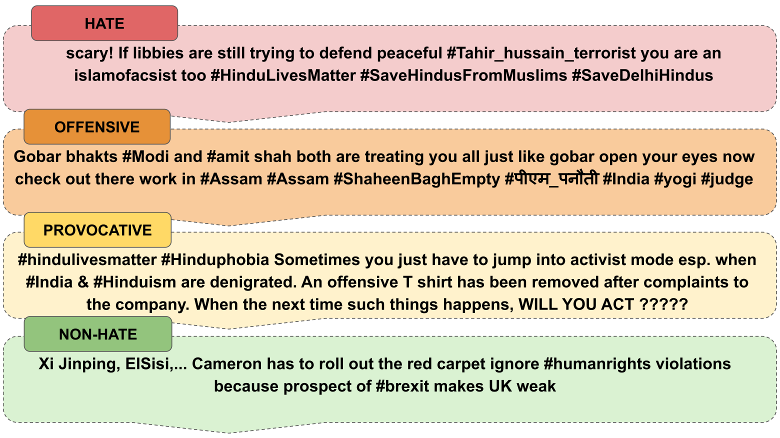


Figure 3.16: Examples (verbatim) of hateful, offensive, provocative, and non-hateful samples from `GOTHate`.

Annotated samples. Figure 3.16 illustrates representative samples of `GOTHate` obtained from crowdsourced annotations. The hateful example critically targets the liberals and the Muslim community in India. However, the tweet's indirect language and requirement of knowledge about 'Islamophobia' make it difficult to mark it as hate. The offensive statement targeting 'Narendra Modi' and 'Amit Shah' makes a demeaning comparison between 'Gobar' (cow dung) and 'Gobar Bhakts' (dung followers) without using any explicit swearwords. This post again requires information about Indian politics for correct annotation. Meanwhile, the provocative example invokes a call to action in the context of #HinduLivesMatter. Lastly, the benign example hints at a subtle mockery of 'Xi Jinping' but is not harsh enough to mark it as either hateful, offensive, or provocative. In short, `GOTHate` appears to be challenging for hate speech classification models.

Takeaways from data curation. Succinctly speaking, our data collection and annotation exercise shows that:

– Crowdsourcing comes at the cost of expertise in terms of IAA (0.71 vs 0.80).
– The overall agreement scores reinstate the difficulty in annotating hate speech.
– 60% of all disagreements are observed in the provocative class. The difference stems from the seemingly neutral phrasing of provocation.

> Based on our extensive data curation exercise, we can safely attest that practitioners dealing with hate speech annotations need to evaluate the cost, quality, and diversity of annotations while developing the schema.

3.4.4 Comparison of GOTHate with existing hate speech datasets

Given the cornucopia of hate speech datasets, a comparison of GOTHate and the existing benchmarks is imperative. Borrowing from Section 3.3.2, we employ Davidson (Davidson *et al.*, 2017), Founta (Founta *et al.*, 2018), HASOC19 (Mandl *et al.*, 2019), OLID (Zampieri *et al.*, 2019*a*), and HatEval (Basile *et al.*, 2019). We also compare GOTHate with LatentHatred which is an implicit hate corpus (ElSherief *et al.*, 2021). Of these datasets, only HatEval is neutrally seeded. For our current analysis, we omit the neutrally seeded dataset from Stormfront (de Gibert *et al.*, 2018) as it is an unmoderated platform. In terms of code-mixed and Indian topics, HASOC19 and our previously proposed ConInHate (Masud *et al.*, 2021) datasets come closest. The former is in code-mixed Hindi, and the latter is in English. For the LatentHatred dataset, we combine the classes of implicit and explicit hate into a single hate category. For the rest of the datasets, the label distribution stays intact. An outline of various datasets is provided in Table 3.13. Class labels Hate (H), Offensive (O), Provocative (P), Neutral (N), Abuse (A), and Spam (S) cover all labels in our dataset and the benchmarks.

Table 3.13: The number of samples, unique labels, and the percentage of hateful samples in the datasets used for comparison.

Dataset	# Samples	# Labels	Hate%
GOTHate	51367	4 (H,O,P,N)	7.23
Founta	59189	4 (H,A,S,N)	4.05
Davidson	24783	3 (H,O,N)	5.77
HASOC19	17657	2 (H,N)	34.92
OLID	10592	2 (O,N)	32.91
HatEval	19600	2 (H,N)	41.88
LatentHatred	21480	2 (H,N)	38.12
ConInHate	23748	2 (H,N)	4.68

Experimental setup. For experiments in this section, we employ two classification models. The first one is an n-gram ({1,2,3}) based TF-IDF Logistic Regression model (hereafter referred to as TF-IDF). The second is a simple CH trained on the CLS token obtained from a frozen mBERT (hereafter referred to as mBERT). To account for the varying positive and negative class sizes, we also report the Matthews correlation coefficient (MCC) along with macro-F1 to establish the lower correlation of our dataset's prediction. MCC ranges from +1 to -1, with +1 being the best agreement between predicted and expected values, and 0 means random prediction.

Data drift analysis. Among the existing benchmarks, we study how distinguished their lexical and semantic spaces are in relation to GOTHate. Following the idea of data drift analysing datasets for PLMs in Section 3.3.2, we once again employ data drift (Lu *et al.*, 2019) to capture the dissimilarity in feature space between GOTHate and other benchmarks, with GOTHate being the target dataset.

Under both models, we observe similar patterns. The lowest results, i.e., highest latent similarity, are obtained from HASOC19, a code-mixed Hinglish corpus. For both TF-IDF and mBERT models, we get 0.93 (0.87) and 0.81 (0.96) macro-F1 (MCC). Yet, Table 3.14 highlights that none of the dataset comparisons, including HASOC19 has an F1 <$.5$ or MCC ≈ 0, which would have indicated that the feature space of GOTHate is indistinguishable from existing datasets. This corroborates the variability in feature space captured by GOTHate. It is interesting to note that for some datasets, such as OLID and HatEval the results for TF-IDF and mBERT are not proportional. Meanwhile,

Table 3.14: Data drift with `GOTHate` as the target dataset against existing datasets as the source. We report macro-F1 (F1) and Matthews correlation coefficient (MCC). The lower the F1, the closer the source and target datasets.

Source Dataset	TF-IDF		mBERT	
	F1 ↓	MCC	F1↓	MCC
`Founta`	0.98	0.95	0.94	0.86
`Davidson`	0.99	0.97	0.98	0.96
`HASOC19`	**0.93**	0.87	**0.81**	0.96
`OLID`	0.97	0.91	0.91	0.89
`HatEval`	0.99	0.97	0.89	0.81
`LatentHatred`	0.98	0.95	0.98	0.33
`ConInHate`	0.96	0.93	0.99	0.99

TF-IDF `ConInHate` shows the expected drop in macro-F1 but does not under mBERT. We hypothesise these deviations to arise from the anisotropic behaviour that BERT-based embeddings have been recorded to exhibit (Ethayarajh, 2019).

Inter-class similarity. Besides the comparison among datasets, we are also intrigued to observe at a granular level the similarity in the latent space among the various classes of a given dataset. We hypothesise a relatively higher inter-class similarity among the classes in `GOTHate` compared to its counterparts. To capture the inter-class proximity within a dataset, we carry out the analysis under two setups:

1. **Lexical.** For each dataset, we generate a Laplacian smoothed unigram distribution of each class and employ Jensen–Shannon (JS) divergence (Nielsen, 2019) to compare inter-class distributions within a dataset. The lower the JS, the harder it is to separate the classes.

2. **Semantic.** For each data point in a class, the semantic embedding vectors are obtained from the CLS token of an mBERT. We employ maximum mean discrepancy (MMD) (Gretton *et al.*, 2012) to establish similarity. For each dataset, we report the median (mean) MMD over 100 runs with 30 random pairs (label-wise) of samples (without replacement).

Analogous to the adversarial validation examined before, from Table 3.15, we once again observe a lower variability in terms of semantic similarity across datasets and classes. Overall, in terms of both lexical and semantic similarities, all hate speech datasets register relatively lower inter-class divergence. It corroborates the difficulty in annotating and modelling subjective tasks like hate speech. We hypothesise that size, source, nature, and topic of collection all play a role in inter-class similarity, and one needs to examine the datasets, controlling one aspect at a time, to quantify their influence on the curated dataset. In `GOTHate` the reason for the further lowering of divergence can be attributed to the topical similarity and use of neutral seeding for data collection. In terms of hate-neutral lexical distributions, `GOTHate` has a lower JS divergence (0.12) than the more explicit counterparts of `Founta` (0.34) and `Davidson` (0.32). Additionally, within `GOTHate`, the JS divergence values for the pairs of H-P=0.09 and N-P=0.06 are lower than those of other inter-dataset pairs. This low divergence is a cause for the high disagreement in the provocative class, as well as the underlying reason for it.

Table 3.15: Inter-class similarity within a dataset measured in terms of Jensen–Shannon divergence (JS) and maximum mean discrepancy (MMD). JS is obtained via n-grams. MMD is obtained via mBERT embeddings. For MMD, we report the mean and standard deviation of over 100 runs. For JS and MMD, the lower the scores, the higher the similarity.

Dataset	Label		JS↓	MMD↓	Dataset	Label		JS↓	MMD↓
GOTHate	H	O	0.14	0.04±0.02	Founta	H	A	0.21	0.02 ±0.02
	H	P	0.09	0.03±0.02		H	S	0.48	0.06±0.05
	H	N	0.12	0.03±0.02		H	N	0.32	0.04±0.03
	O	P	0.12	0.03±0.02		A	S	0.47	0.07±0.06
	O	N	0.12	0.03±0.02		A	N	0.32	0.06±0.05
	P	N	0.06	0.02±0.01		S	N	0.24	0.04±0.03
Davidson	H	O	0.20	0.04±0.03	HASOC19	H	N	0.25	0.12±0.10
	H	N	0.35	0.04±0.03	OLID	O	N	0.12	0.04±0.02
	O	N	0.33	0.07±0.05	HatEval	H	N	0.13	0.04±0.04
LatentHatred	H	N	0.12	0.04 ±0.03	ConInHate	H	N	0.25	0.04 ±0.03

3.4.5 How hard is GOTHate to classify?

Based on the comparisons with other hate speech datasets as well as the lower inter-class divergence of our dataset, we hypothesise that GOTHate will be challenging to classify based on the text of the incoming post alone. Employing the same experimental setup as in Section 3.4.4, we record overall macro-F1 and MCC for each dataset.

Table 3.16: Overall hate speech classification performance of hate speech datasets in terms of macro-F1 and MCC via TF-IDF and mBERT-based modelling. The higher the F1, the better classified the dataset.

Dataset	TF-IDF		mBERT	
	F1↑	MCC	F1↑	MCC
GOTHate	0.39	0.22	0.28	0.14
Founta	0.55	0.46	0.51	0.55
Davidson	0.66	0.59	0.51	0.49
HASOC19	0.68	0.38	0.71	0.43
OLID	0.66	0.32	0.62	0.29
HatEval	0.57	0.22	0.58	0.18
LatentHatred	0.67	0.33	0.66	0.33
ConInHate	0.78	0.57	0.99	0.99

As evident from Table 3.16, GOTHate registers the lowest F1 under both modelling setups. In terms of TF-IDF, Founta (a dataset with similar size and label granularity) scores a macro-F1 of 0.55 compared to 0.39 of GOTHate. Meanwhile, the code-mixed binary dataset HASOC19 reports an even higher performance of 0.68. Consequently, looking at mBERT-based performances GOTHate reports a macro-F1 of 0.28. In comparison, Founta and HASOC19 score a macro-F1 of 0.51 and 0.71, respectively. Further, in terms of neutral seeding HatEval reports a macro-F1 of 0.58. It can, therefore, be concluded that in the absence of additional signals, a dataset such as GOTHate encompasses the diversity across languages, geographies, and topics, and is hard to model (Schmidt and Wiegand, 2017) based solely on the text of the post. The issue is exacerbated in the case of fine-grained hate speech classification, such as four labels in our dataset, wherein the already indistinct label boundaries (Alkomah and Ma, 2022) get even blurrier. *The results motivate us to combine context signals inspired by our prior work (Masud et al., 2021) (Section 3.2). Further, the mBERT in a frozen state* *seems to be providing a limited advantage. This calls for an examination of strategy borrowing from our prior work (Masud* et al.*, 2024b) (Section 3.3).*

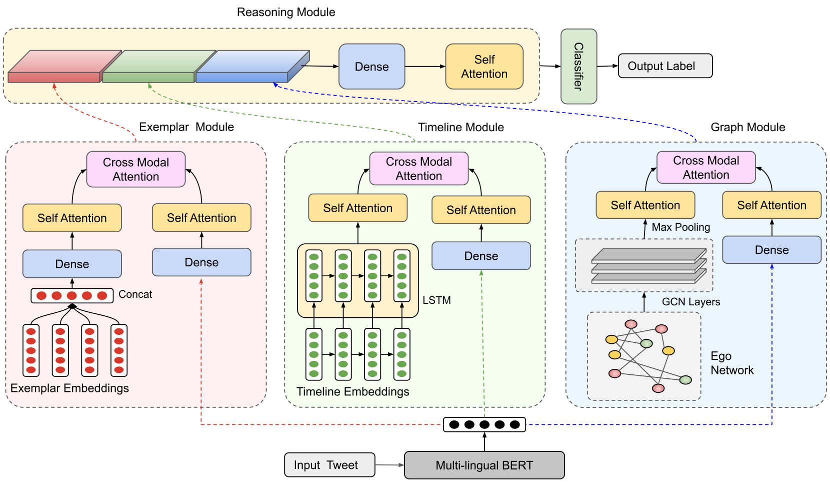


Figure 3.17: Model architecture of `HEN−mBERT`. The embedding for an incoming tweet is obtained from mBERT. The exemplar and timeline modules obtain the input embedding in the form of the mBERT `CLS` token. Meanwhile, the graph module receives input from the corresponding user's ego network. Each module enhances the respective input embedding via cross-modal attention. Ultimately, the reasoning module concatenates the three endogenous signals to obtain a context-rich embedding, which passes through an attentive feed-forward block for classification.

3.4.6 Design of `HEN−mBERT` for contextual hate detection

Our analysis of `GOTHate` and its performance on a simple setup exposes a performance gap. This gap allows us to revisit the findings from both incorporating contextual signals and PLM finetuning strategies to design a context-rich hate speech detection system. Borrowing from the exogenous cross-attention framework (Section 3.2), we explore the notion of endogenous cross-attention. To this end, we propose `HEN−mBERT`, a modular mixture-of-experts PLM finetuning setup that enriches the textual representations with the ancillary signals. Figure 3.17 outlines the architecture of `HEN−mBERT`. Attention-based infusion brings the latent subspaces of different auxiliary signals closer to the textual subspace of the incoming post.

Base LM. We employ mBERT (Devlin *et al.*, 2019) as our base PLM and use the `CLS` token of the last layer to generate embeddings for the exemplar and timeline modules. For the interaction module, we utilise the representations from the last layer of mBERT given by $Z \in \mathbb{R}^{l \times d}$, where d is the feature dimension of PLM, and l is the maximum sequence length.

Reasoning module. The reasoning module aptly merges the information from each signal via mutual interaction. We obtain the context-rich vectors $\hat{Z}_g$, $\hat{Z}_e$, and $\hat{Z}_t$ from the exemplar, timeline and graph modules respectively. We concatenate the vectors and apply a non-linear transformation for dimensionality reduction, resulting in vector $Z_{final} \in \mathbb{R}^{l \times d}$. This vector is further enhanced using a multi-headed self-attention mechanism. The final vector is then passed through a simple CH.

Exemplar module

Objective. Exemplars refer to the set of examples from the training set that are semantically related to the incoming post. They help provide stylistic and thematic cues to improve class discrimination (Schmaltz, 2021). Further, exemplars can be obtained from within the dataset without requiring additional data curation. We hypothesise that providing examples can compensate for the higher disagreement in the provocative class.

Formulation. We start by augmenting this signal using dense retrieval via mSBERT-based setup (Reimers and Gurevych, 2019). In a preliminary analysis, we also experimented with BM25 (Robertson and Zaragoza, 2009), but mSBERT fared better. We employ a label-based exemplar search for the training set, where the exemplars are retrieved from the same label subspace. We extract label-invariant (without knowing the label for an incoming post) exemplars for the validation and test sets using the training dataset for grounding. For each instance, we select the top exemplars based on cosine similarity. We analyse some exemplars as a part of the error analysis in Section 3.4.9.

We concatenate the exemplars $P = \{p_{i1}, ..., p_{ik}\}$ as $P_e \in \mathbb{R}^{k \times d}$. It is followed by a non-linear transformation and dimensionality reduction, casting P_e to $F_e \in \mathbb{R}^{k \times d_f}$. Consequently, the base language vector $Z \in \mathbb{R}^{l \times d}$ is reduced to $Z_e \in \mathbb{R}^{l \times d_f}$. To extract salient features from F_e and Z_e, they are independently passed to a self-attention module (Vaswani *et al.*, 2017) to generate module-specific F'_e (Equation 3.5) and Z'_e (Equation 3.6), respectively. To facilitate semantic interaction between the exemplars and incoming posts, we generate the exemplar-infused language representation $\hat{Z}_e$ (Equation 3.7). The enriched vectors interact with each other via multi-headed cross-modal attention.

$$F'_e = Softmax\left(\frac{F_e F_e^T}{\sqrt{d_f}}\right) F_e \quad (3.5) \qquad Z'_e = Softmax\left(\frac{Z_e Z_e^T}{\sqrt{d_f}}\right) Z_e \quad (3.6)$$

$$\hat{Z}_e = MultiHead\left(Softmax\left(\frac{F'_e Z'^T_e}{\sqrt{d_f}}\right) Z'_e\right) \quad (3.7)$$

Timeline module

Objective. While examples help obtain latent signals from within the dataset, users' propensity for posting hateful content is not just a one-time incident. Existing literature, as well as our experience with `ConInHate` (Masud *et al.*, 2021), corroborates that users' historical data provides crucial insights into whether they will post hateful material in the future (Qian *et al.*, 2018; Liu *et al.*, 2024; Noorian *et al.*, 2024). We showcase an example of historical influence as a part of the error analysis in Section 3.4.9.

Formulation. We concatenate historical posts of a user, $T_i = \{t_{i1}, ..., t_{ia}\}$, as $F_t \in \mathbb{R}^{a \times d}$. The sequential and relatively temporal nature of timeline information is further encapsulated by two layers of LSTM, generating temporally-enriched vector $\hat{F}_t$. Here as well, Z is reduced to $Z_t \in \mathbb{R}^{l \times d_f}$. To extract the temporal and language-specific nuances, the vectors $\hat{F}_t$ and Z_t undergo self-attention operations via Equations 3.8 and 3.9. This results in their enriched form of vectors given by F'_t and Z'_t. Finally, the historical information is infused via the multi-headed cross-modal attention in Equation 3.10, generating the output of the timeline module as $\hat{Z}_t$.

$$F'_t = Softmax\left(\frac{F_t F_t^T}{\sqrt{d_f}}\right) F_t \quad (3.8) \qquad Z'_t = Softmax\left(\frac{Z_t Z_t^T}{\sqrt{d_f}}\right) Z_t \quad (3.9)$$

$$\hat{Z}_t = MultiHead\left(Softmax\left(\frac{F'_t Z_t'^T}{\sqrt{d_f}}\right) Z'_t\right) \quad (3.10)$$

Graph module

Objective. One must also account for the network effect as the information a user eventually interacts with is mainly obtained from their friend circle on the platform (Mathew *et al.*, 2020). As hateful users are likely to follow and retweet other toxic users (Ribeiro *et al.*, 2018), we also examine the role of the ego network in our setup.

Formulation. We begin by constructing a directed homogeneous network of all the root users, their first-hop followers/followees, and the users retweeting the root users' tweets. To prune the network, we select the top 100 followers, followee, and retweeters with the highest node centrality for each user. Then, we add edge weights to distinguish the different interaction types. Initially, all the edges are given a weight of 1, except the self-loop weight of 0.1. Followers/ followees who have at least once retweeted the root user earn higher precedence with an edge weight of 1.5. Meanwhile, external retweeters not included in the root users' follower-followee are given slightly less weight of 0.5.

For a post, x_i by a user u_j, we define a directed and weighted user-level ego network as $o_j = G(V'_j, E'_j)$. Each node in V'_j is initialised with node embeddings of dimension d_g. The initial node embeddings are generated with node2vec (Grover and Leskovec, 2016). In the preliminary analysis, we also experimented with GraphSage (Hamilton *et al.*, 2017), but node2vec fared better. The ego network is processed via a three-layered graph convolution (Kipf and Welling, 2017) followed by a max-pooling to spawn an aggregated ego network embedding of the user $F_g \in \mathbb{R}^{d_f}$, where $d > d_f > d_g$ (Equation 3.11). The language vector Z is once again reduced to $Z_g \in \mathbb{R}^{l \times d_f}$. F'_g and Z'_g are generated to obtain the independent self-attention w.r.t graph module via Equations 3.12 and 3.13. Finally, we generate the network-aware language representation $\hat{Z}_g$ (Equation 3.14) via multi-headed cross-modal attention.

$$F_g = GraphConv(o_j)) \quad (3.11)$$

$$F'_g = Softmax\left(\frac{F_g F_g^T}{\sqrt{d_f}}\right) F_t \quad (3.12) \qquad Z'_g = Softmax\left(\frac{Z_g Z_g^T}{\sqrt{d_f}}\right) Z_t \quad (3.13)$$

$$\hat{Z}_g = MultiHead\left(Softmax\left(\frac{F'_g Z_g'^T}{\sqrt{d_f}}\right) Z'_g\right) \quad (3.14)$$

3.4.7 Experimental setup

Dataset splits. `GOTHate` samples are split into 80-10-10 ratios stratified over the label for train, val and test sets, respectively.

Hyperparameters. For the statistical models, we use word TF-IDF features using scikit-learn. Meanwhile, CNN, LSTM, and Founta baselines incorporate Glove EN+HI (English + Hindi) embeddings (Pennington *et al.*, 2014) to account for the code-mixed nature of the data. The transformer-based models of mBERT, ARHNet, HurtBERT, and `HEN-mBERT` are trained using the Huggingface (Wolf *et al.*, 2020) in PyTorch. After manual hyperparameter tuning, we employ the following values in the final model: learning rate: $2e-5$, weight decay: $1e-4$, batch size: 32, max epochs: 20, optimiser: Adam (Kingma and Ba, 2014). We select the best validation model after the training macro-F1 crosses 70%. All the codes are seeded to a value of 42. VIDIA RTX A6000 of 48GB is used for all the experiments. All the transformer-based models occupy around 19GB of GPU and require approximately $6-8$ minutes per epoch.

Baselines. In order to establish the efficacy of `HEN-mBERT` we compare it against several baselines divided into traditional, neural, and transformer-based, all trained and tested with `GOTHate`.

- **Traditional baselines.** We employ Naive Bayes (NB), Logistic Regression (LR), and Support Vector Machines (SVM) based classifiers with n-gram based TF-IDF based features. The Davidson model (Davidson *et al.*, 2017) serves as a text feature-rich version of the LR model. All traditional baselines employ only the textual features of the incoming root tweet.
- **Neural baselines.** In the next set of models, we experiment with vanilla CNN (Kim, 2014) and LSTM (Schmidhuber, 1997) with concatenated Glove embeddings (EN+HI) as input. The vanilla models also employ only the textual features. The Founta (Founta *et al.*, 2019) baseline incorporates the notions of self-attention-based in the RNN method, again given Glove (EN+HI) embedding of the post as input. Additionally, it concatenates numerical metadata such as the number of followers, followed by the user, and the number of engagement metrics.
- **Transformer baselines.** We begin by experimenting with mBERT only on the textual representation of the root tweet. mBERT also serves as the base model for our endogenous signal infusion. Additionally, we train two existing hate detection models that incorporate external signals. ARHNet (Ghosh Chowdhury *et al.*, 2019) concatenates the language representations with node2vec user embeddings as a late fusion combination. Meanwhile, HurtBERT (Koufakou *et al.*, 2020) jointly encodes multilingual hate lexicon knowledge with the language representations. For comparison, we employ mBERT as the base embedding for both these baselines instead of the original BiLSTM in ARHNet and BERT in HurtBERT.

Evaluation metrics. At a fine-grained level, we report the class-wise and overall precision (P), recall (R), and macro-F1 (F1).

Ablations. We compare the performance of encoding only a single endogenous signal, leading to three ablation models.

3.4.8 Results

Tables 3.17 and 3.18 encapsulate the comparative performance of `HEN-mBERT`.

Traditional baselines. The traditional machine learning baselines (M1-M3) in Table 3.18 yield inferior performances on hate and provocation labels. Upon granular

Table 3.17: Performance analysis of `HEN-mBERT` and competing models. We report class-wise and overall precision (P), recall (R), and macro-F1 (F1).

Model	Hate			Offensive			Provocative			Neutral			Overall		
	P	R	F1	P	R	F1	P	R	F1	P	R	F1	P	R	F1
M1: NB	0.4468	0.1129	0.1802	0.4027	0.4287	0.4153	0.4253	0.2322	0.3004	0.6557	0.8150	0.7267	0.4826	0.3972	0.4057
M2: LR	**0.4646**	0.1239	0.1957	0.5714	0.3618	0.4430	0.4764	0.2490	0.3271	0.6576	0.9004	**0.7601**	**0.5425**	0.4088	0.4315
M3: SVC	0.4357	0.1639	0.2382	0.5192	0.4002	0.4520	0.4072	0.3099	0.3519	0.6690	0.8232	0.7382	0.5078	0.4243	0.4451
M4: Davidson	0.1732	0.4516	0.2504	0.3814	0.4362	0.4070	0.3472	0.4038	0.3734	0.7402	0.5143	0.6069	0.4105	0.4515	0.4094
M5: CNN	0.1824	0.3844	0.2474	0.3307	**0.5569**	0.4148	0.3593	0.3848	0.3716	0.7683	0.4936	0.6019	0.4101	0.4548	0.4087
M6: LSTM	0.2083	0.3387	0.2579	0.4168	0.4907	0.4508	0.3404	0.5005	0.4052	**0.7715**	0.5398	0.6352	0.4347	0.4674	0.4373
M7: Founta	0.1345	**0.4543**	0.2075	0.3392	0.3841	0.3603	0.2925	0.2806	0.2864	0.7022	0.4726	0.5650	0.3671	0.3979	0.3548
M8: mBERT	0.3309	0.2419	0.2795	0.4884	0.4994	0.4939	**0.3964**	0.3754	0.3856	0.7119	**0.7458**	0.7285	0.4819	0.4656	0.4719
M9: ARHNet	0.2770	0.2956	0.2860	0.5056	0.4423	0.4719	0.3729	**0.4492**	**0.4075**	0.7321	0.6968	0.7140	0.4719	0.4710	0.4699
M10: HurtBERT	0.2836	0.2607	0.2717	**0.5266**	0.4646	0.4937	0.3751	0.4341	0.4024	0.7676	0.7092	0.7143	0.7194	0.4672	0.4705
M11: `HEN-mBERT`	0.3698	0.3091	**0.3367**	0.5103	0.4895	**0.4997**	0.3852	0.4294	0.4061	0.7392	0.7323	0.7358	0.5011	**0.4901**	**0.4946**

Table 3.18: Performance summary of `HEN-mBERT` and baselines in terms of macro-F1. We also report the performance of `HEN-mBERT`'s ablation setups.

Model	Hate	Offensive	Provocative	Neutral	Overall
M1: NB	0.1802	0.4153	0.3004	0.7267	0.4057
M2: LR	0.1957	0.4430	0.3271	**0.7601**	0.4315
M3: SVC	0.2382	0.4520	0.3519	0.7382	0.4451
M4: Davidson	0.2504	0.4070	0.3734	0.6069	0.4094
M5: CNN	0.2474	0.4148	0.3716	0.6019	0.4087
M6: LSTM	0.2579	0.4508	0.4052	0.6352	0.4373
M7: Founta	0.2075	0.3603	0.2864	0.5650	0.3548
M8: mBERT	0.2795	0.4939	0.3856	0.7285	0.4719
M9: ARHNet	0.2860	0.4719	0.4075	0.7140	0.4699
M10: HurtBERT	0.2717	0.4937	0.4024	0.7143	0.4705
M11: `HEN-mBERT`$_E$	0.3090	0.5019	**0.4395**	0.6977	0.4870
M12: `HEN-mBERT`$_T$	0.2894	0.4853	0.4271	0.6987	0.4751
M13: `HEN-mBERT`$_G$	0.2829	**0.5189**	0.3891	0.6938	0.4712
M14: `HEN-mBERT`	**0.3367**	0.4997	0.4061	0.7358	**0.4946**

examination of Table 3.17, we observe that these systems are characterised by a high recall for the benign samples and a low recall for the hateful ones, indicating that such models are aggressive in predicting samples as not hateful. By augmenting the n-gram features with semantic features (pos-tags) and textual meta-data (Vader score, hashtag counter, etc.), the Davidson (Davidson *et al.*, 2017) model (M4) slightly improves the performance for the hate class, compared to vanilla statistical baselines. However, the skewness of the class labels and over-dependency on word co-occurrence contribute to the downfall of these systems.

Neural baselines. We observe that CNN and LSTM neural models (M5-M7) fare better than the traditional ML ones, but still perform poorly. They do register about $2-5\%$ gain over conventional methods for the hate and provocative instances in terms of the F1 score. However, the macro-F1 for offence remains more or less the same.

Transformer baselines. As expected, by finetuning a PLM-based setup, even without any contextual signal, we see a substantial gain over traditional and neural baselines. By applying mBERT (M8) with only input post as a feature– a 2%, 4%, 3%, and 4% improvement in macro-F1 for hate, offensive, provocative, and non-hate labels, respectively, is evident. Though we see about 1% improvement for the hated class, overall, ARHNet does not fare better than naive mBERT. On the other hand, the HurtBERT model does not showcase any improvement over mBERT. From ARHNet, we can see that it is not only the inclusion of contextual signals but also how they are infused that impacts the performance of hate speech detection. In the case of HurtBERT, the performance loss can be attributed to dependency on hate lexicons, which do not align well with neutrally seeded datasets such as `GOTHate`.

Proposed method. `HEN-mBERT` and its variants produce a balanced performance, especially for the harder-to-classify classes of hate and provocation. The individual latent augmentations are enlisted as $\texttt{HEN-mBERT}_{\{E, T, G\}}$ to represent the addition of only exemplar, historical, or network-based features, respectively. As shown in Table 3.18, the performance for hate classification improves by adding each module. The exemplar-only module (M11: $\texttt{HEN-mBERT}_{E}$) reports an overall increase in performance over mBERT. Infusing only the exemplar signals improves hate classification by 2.4 macro-F1. We conjecture that adding exemplary tweets helps the model unravel the stylistic and semantic nuances across these labels. This is also true for the provocative class, which witnesses the highest disagreement during manual annotation. While existing baselines reached macro-F1 of 0.41 for provocation, $\texttt{HEN-mBERT}_{E}$ improves it to 0.44. The inclusion of the timeline module (M12: $\texttt{HEN-mBERT}_{T}$) also provides an improvement over existing baselines. However, the results do not improve over the exemplar setup. On the other hand, the graph module (M13: $\texttt{HEN-mBERT}_{G}$) scores the highest for the offensive class. Finally, combining all three signals via `HEN-mBERT` (M14) leads to an improvement in performance across all the classes, with a significant jump of about 5% in the hate label vs the mBERT. The troublesome, provocative class also enjoys a 2% rise in the macro-F1. Combining all three variants also keeps the neutral class's performance on the higher side (0.74). Interestingly, we observe that the most recent timelines, as small as +/-5, combined with a slightly stale but weighted network of most recent interactions, as infused in HEN-mBERT, lead to a 6-point increase in macro-F1 of hate class (compared to mBERT).

3.4.9 Error analysis

We begin our error analysis by looking into some misclassified data points and the role of timeline information. Our label-wise analysis also sheds light on the bias in annotation for subjective tasks such as hate speech. We then perform a manual assessment of randomly sampled exemplars. We conclude our discussion by evaluating the root users in our dataset and commenting on the complexity of the proposed setup.

Label prediction. Given that `HEN-mBERT` extends the mBERT architecture, it is critical to compare the quality of their predictions. While `HEN-mBERT` is enriched with signals to provide additional context, mBERT relies only on the information captured within a post's text. For tweets that contain explicit offences such as "APP is ISIS OF INDIA" and "Arrest Sonia Gandhi" (#1 in Table 3.19), both the models effectively pick up on the cues and make correct predictions. However, in examples that required contextual knowledge (#2 in Table 3.19), mBERT falters. A closer analysis of the root user of this tweet reveals that they actively post Islamophobic content. Consider the following sample tweets from the user's timeline, collected from before (t_b) and after (t_a) the tweet under examination: ($\mathbf{t_b}$) "*look at what Hindus living in mixed-population localities are facing, what $MENTION$ had to face for merely asking his Muslim neighbours not to harass his daughter sexually...and even then if you ask why people don't rent to Muslims, get your head examined.*" ($\mathbf{t_a}$) "*$MENTION$ and $MENTION$ naah...Islamists will never accept Muslim refugees; they will tell the Muslims to create havoc in their home countries and do whatever it takes to convert Dar-ul-Harb into Dar-ul-Islam. Something we should seriously consider doing with Pak Hindus too*". Using such information, our model develops an understanding of this user's hate propensity.

Lastly, given the subjectivity involved in discussing hate, one must also consider

Table 3.19: Error analysis of mBERT and HEN−mBERT analysing examples of *correct classification (#1), misclassification (#2), and mislabelling (#3).*

Test tweet	Gold	mBERT	HEN−mBERT
Example 1: $MENTION$ $MENTION$ Arvind Kejriwal's AAP is ISIS OF INDIA. HIS PROPAGANDA WILL DESTROY HINDUS FROM INDIA WITH HELP OF CONGRESS AND LEFTIST. AVE HINDU SAVE BHARAT. SAVE BRAHMIN, SAVE DALITS, SAVE HINDU OTHER CAST delhi burns delhi riots2020 arrest sonia gandhi delhi violence	O	O	O
Example 2: this must have been a apka tahir offer to jihadis (terrorist) - kill kaffirs, loot their property, do what u want with any kaffir (non-believer) female that "ur right hand possess" ...in short, maal-e-ganimat delhi riots delhi riots	H	P	H
Example 3: $MENTION$ It's not the number, it's the % increase and the doubling time. This administration is just not up to it. They thought the exit would be easy and were wrong, and they are in grave danger of causing much misery through vacillation and poor choices. ovid19 $URL	P	P	O

the case of mislabeled annotations. As observed from #3 in Table 3.19, though the tweet seems innocuous and should be labelled non-hate based on the post's content, it is annotated as provocative and predicted the same by mBERT, while HEN−mBERT predicts it as offensive. Such mislabelling indicated that no hate speech pipeline is immune to annotation biases.

Sample exemplars. Inspired by the positive impact of timeline information in example #1 in Table 3.19, we also evaluate the other textual signal, the exemplars. We randomly sample exemplars from the training and validation sets listed (verbatim) below and comment on their quality. The first example is from the training set, and the other two are from the validation set. Thus, for the first example, the exemplars come from the nearest samples within the same class/label. For the validation examples, the closest samples can be from other classes as well, which reflects the online mode where the labels may not have been known a priori.

#1 Root post's label - Offensive "*$MENTION$ DERANGED DELUSIONAL DUMB DICTATOR DONALD IS MENTALLY UNSTABLE! I WILL NEVER VOTE REPUBLICAN AGAIN IF THEY DON'T STAND UP TO THIS TYRENT LIVING IN THE WHITE HOUSE! k republicans worst dictator ever unstable dictator URL.*"

- **Extracted post's label - Offensive** *"$MENTION$ COULD WALK ON WATER AND THE never trump WILL CRAP ON EVERYTHING HE DOES. SHAME IN THEM. UNFOLLOW ALL OF THEM, PLEASE!"*
- **Extracted post's label - Offensive** *"$MENTION$ $MENTION$ $MENTION$ AND Remember president loco SAID MEXICO WILL PAY FUC**kfu ck trump f*** gop f*** republicans Make go fund me FOR HEALTH CARE, COLLEGE EDUCATION, CLIMATE CHANGE, SOMETHING GOOD AND POSITIVE !! Not for a f**ing wall go fund the wall the resistance resist URL"*

#2 Root post's label - Provocative *"Where is Award Wapsi (return) gang? Are they hiding behind $MENTION$ Lungi (dress)? Hindu lives matter"*

- **Extracted post's label - Provocative** *"Where's the outrage? Where's the Musalman-Khatre-Mein-hai (Muslims-are-in-danger) Gang? Where's the Bhim army now? Pathetic cronies. Hindu lives matter URL"*
- **Extracted post's label - Provocative**: *"$MENTION$ Where are the Bollywood liberals ? Where is their condemnation and outrage? Hindu lives matter"*

#3 Root post's label - Hate *"Ankit Sharma. Gathered intelligence to protect this nation. In this nation, he is dragged by 40-50 people, undressed, tortured, face* *and upper body smashed with heavy objects till his last breath. This is my nation. delhi riots2020 arrest tahir hussain URL"*

- **Extracted post's label - Hate** *"Appalled by the targeted killing and violence against Muslims in Delhi. With so many lives lost, houses burned, and places of worship torched, we request the Government of India to restore normalcy ASAP. Delhi riots delhi riots2020 URL"*
- **Extracted post's label - Provocative** *"The son of Muddasir Khan, one of the martyrs of the anti-Muslim Delhi riots, weeps over his innocent father's body. We are keeping an account. We won't forget anything. ANYTHING AT ALL! Delhi riots facism Indian muslims in danger pic.URL"*

The example $\#1$ from the above list is from the training data. We observe that the retrieved exemplars are of high quality as they adopt a label-based exemplar search for the training samples. We then query exemplars from the training dataset without knowing the label for the validation and test sets, as they are supposed to be unseen. For $\#2$, we see that the model retrieves apt exemplars that share the same theme, sentiment, and label as the root tweet. For example, $\#3$, on the other hand, retrieved exemplars share topical similarity with the root tweet but sometimes miss the actual context and the label connotations.

Overlap in users. As evaluating the quality of the network in terms of hate speech is not directly feasible, we assess the root users in terms of class overlap instead. We look into the overlap in terms of users covered in the respective classes among the annotated and predicted test samples. 354 unique users are present in the hateful samples of the test set's ground truth annotations. Meanwhile, from among the posts labelled as hate speech via `HEN-mBERT` we get a set of 293 users, an overlap of 158 users with the 354 ground users. Comparing this number with the LR (M2) model, which scores the highest macro-F1 for the neutral class, we observe that LR predicts only 86 users as hateful, with an overlap of only 48. Owing to its tendency to consider everything neutral, LR possesses comparatively lower user overlap.

Consequently, in terms of unique users, our model gives more false positives for hate (174) than the LR model (38). This does not necessarily imply that our model is overfitting. The proposed model is relatively more aggressive in predicting hate than the LR model. In the real world, we would prefer that both false positives and false negatives be low. However, it is better to be aggressive in the first level of filtering, as the human moderators can always flag the false positives in the final report. However, if the model defaults to predicting non-hate, then it exposes at-risk groups to a higher volume of hateful content. This difference in user overlap also substantiates the imbalanced precision and recall performances of traditional models recorded in Table 3.17. By carrying a more balanced P-R per class, our proposed (`HEN-mBERT`) set ensures that no group/class is penalised more than others. In future iterations, we aim to use this pipeline to generate feedback from moderators and incrementally train our classifier (Qian *et al.*, 2021).

Note on the model complexity. In our preliminary analysis, we experimented with various modelling techniques and observed that simpler but endogenous signal-rich setups worked better. Building upon `HEN-mBERT` provided in this study is the way forward for context-aware hate speech detection. Because not all internet forums can access all auxiliary features, our pluggable modules can be used as needed.

3.4.10 Limitations and future works

- **Real-time integration of signals.** In our current experiments, the auxiliary signals are already curated. However, to contribute to content moderation in the real world, the system needs to operate online. While the most recent timeline and exemplar information are easier to curate, real-time ego network interactions are not. Furthermore, the coverage of topological information cannot be absolute as the network size expands exponentially as the number of hops increases. Integrating network signals either as an aggregate or incrementally is a possible alternative.
- **Human and model biases.** Given the nature of hate speech, one must acknowledge the annotation biases (Garg *et al.*, 2023) irrespective of the annotation and modelling setup. Our dataset, too, suffers from mislabelling arising from annotator bias and disagreements. We motivate future research to not only account for these disagreements but also employ state-of-the-art tools (LLM prompting in particular) to help improve human annotations.
- **Addressing implicitness beyond provocation.** Despite being a neutrally seeded and fine-grained curation of hatefulness spread over {hate, offence, provocation}, our work does not adequately address the implicit nature of hate. For future work, we press for a more neuro-symbolic approach that facilitates the integration of explanations, commonsense knowledge, and stereotyping motives to uncover implicitness beyond endogenous signals.

3.5 Diffusion visualisation on large networks

3.5.1 Motivation

Research gap. Despite a growing body of work that focuses on mathematical modelling of information diffusion, little work has been done to accommodate visual analysis of diffusion dynamics (Sun *et al.*, 2017; Wu *et al.*, 2014). Recently, there has been tremendous development in analysing epidemics due to the COVID-19 pandemic (Deshmukh *et al.*, 2021; Pan *et al.*, 2020; Yang *et al.*, 2020). However, due to their niche application and platform dependency, the tools cannot be easily exported to work for different settings. In addition, while all major technology companies have access to extensive network data, the visualisation supporting these networks is usually in-house and closed-source. Consequently, there are a limited number of publicly available interfaces to simulate and visualise information diffusion on large networks.

Research questions. Building a proof-of-concept, we seek to answer:

RQ1: What design decisions need to be accounted for in developing a web-based diffusion visualisation tool?

RQ2: How scalable is our system compared to existing diffusion visualisation tools?

Contribution summary. To overcome the shortcomings mentioned above, we introduce **`DiVA`** — an open-source, customised web interface for the study of information diffusion (Sahnan *et al.*, 2022). `DiVA` summarises the diffusion analysis in three forms — (a) statistics, (b) plots, and (c) an interactive network. *The most unique and promising feature of `DiVA` is its ability to visually and numerically compare the simulations of two*

Figure 3.18: Overview of `DiVA` for the primary visualisation mode at iteration $t = 48$. The user uploads a network consisting of $25K$ nodes and $700K$ edges. The user also uploads a custom algorithm (`CustomAlgo.py`) and provides a list of initially infected nodes (`seednodes_2.txt`). The layouts of the six panels visible for this mode are separately marked. The main panel loads the central canvas containing the network. The panels on both sides of the main canvas contain the tools necessary to perform the diffusion analysis. The panels on the top and bottom act as the navigation bar and media control panel, respectively.

diffusion algorithms at the same time. Hence, by extension, it supports simultaneous comparison of the results of a diffusion algorithm against the real-world (a.k.a., ground-truth) diffusion patterns. `DiVA` also supports the user in simulating their custom diffusion algorithms.

3.5.2 Interface overview

`DiVA` offers two visualisation modes — **primary** and **dual**. Additionally, the system offers a wide array of features to analyse the network. A case study of using `DiVA` is provided in Appendix C.

Primary visualisation. By default, this mode is loaded by the system, as illustrated in Figure 3.18. Once a user sets the initial network, `DiVA` displays it in the `main panel`, where a node's colour is graded according to degree centrality. After submitting a diffusion query, the user can graphically view the spread and rate of diffusion. Once the `graph view` is triggered, it dynamically updates the colour of a node as per the node status at the given iteration. The interactive slider interface allows the user to check out the diffusion snapshot (visually and numerically) at a particular timestamp.

Dual visualisation. In `DiVA`, we also introduce an advanced visualisation mode, wherein the user can compare the impact of two diffusion setups (on the same network) in a split view on the `main panel`. We use the same number of iterations and initially infected nodes for a one-to-one mapping of the results per iteration. If a node's status is set as infected by both models for a given iteration, it is counted towards the commonly infected nodes for that iteration. The `report view` now contains three sets of plots:

(a) Diffusion trends per configuration. (b) A plot of the F1-score per iteration. (c) A plot of commonly infected nodes per iteration.

3.5.3 System design

DiVA is primarily implemented using front-end technologies, namely JavaScript and HTML5, and back-end technologies, including Flask and SQLite. The architectural overview of DiVA is provided in Figure 3.19. We use Google's authentication system and file system-based session management. As DiVA is meant to be an online tool, secure authentication and session management are necessary for the system. The structure and layout of the tool are primarily managed with our custom CSS and JavaScript. The front-end and back-end communicate using AJAX and REST APIs.

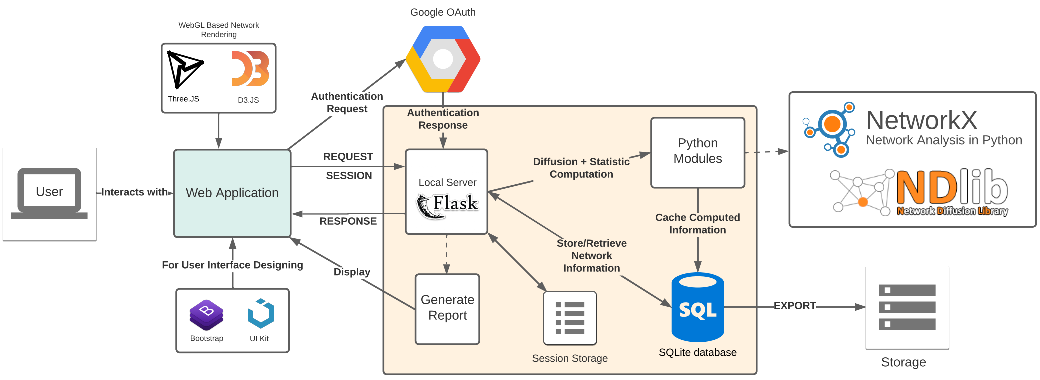


Figure 3.19: Architectural overview of DiVA.

For running diffusion algorithms, we use NDlib, a network diffusion library (Rossetti *et al.*, 2018). Additionally, DiVA supports custom algorithms. Within the Flask server, we use NetworkX (Hagberg *et al.*, 2008) to (a) model the network into an optimised data structure, (b) compute network statistics, and (c) make use of existing high-performance graph representation file formats. We then use a custom network representation file (a JSON-based edge list format) and stream it to the front end asynchronously in chunks through a custom streaming API using JavaScript web workers. The size of the representation file can reach up to 46 MB for a network of 75k nodes and 4.5M edges, which, if transferred in one chunk, hogs the server's resources. Additionally, we provide a mechanism for the user to reuse the computed locations of nodes and edges by saving a custom `.diva` file, which stores a compressed version of the computations run by the user. For visualising the network in the `main panel`, it is fed into d3-force's layout generation algorithm. We then use three.js to plot these nodes and the corresponding edges onto a WebGL context.

3.5.4 Performance comparison

With a prime focus on web interfaces for diffusion visualisation, we examine the scalability challenges of DiVA compared against two other web interfaces built for the same purpose - Epinet and NDlib-Viz. Epinet is a tool built upon the EpiModel (Jenness *et al.*, 2018) package and hosted through Shiny, an R package. NDlib-Viz is a visualisation module built on top of NDlib (Rossetti *et al.*, 2018).

Table 3.20: Performance comparison among three web-based diffusion visualisation tools — `Epinet`, `NDlib-Viz` and `DiVA`, based on their ability to load and run diffusion models for varying graph sizes. ✖ indicates action not supported by the interface. The time to load a random network and run a diffusion model is recorded in seconds as an average of 5 runs. All the systems are evaluated on a locally-hosted Chrome browser (Version 101.0.4951.64) on ThinkPad E480 with 16GB RAM.

# Nodes	**Epinet**		**NDlib-Viz**		**RETINA**	
	Load network	**Run Diffusion**	**Load network**	**Run Diffusion**	**Load network**	**Run Diffusion**
100	1.24	1.36	0.64	1.13	2.28	0.982
300	1.40	1.49	0.694	1.17	2.00	1.16
500	1.39	2.00	1.13	1.08	1.96	1.19
700	1.23	2.18	1.37	1.44	2.14	1.28
1000	1.30	2.47	2.06	2.08	2.69	1.42
3000	1.94	7.47	14.24	6.84	6.72	3.96
6000	6.62	39.02	✖	✖	14.04	9.07
10000	✖	✖	✖	✖	14.57	13.66

Table 3.21: Performance of `DiVA` w.r.t different system parameters. The simulations were run on Google Chrome version 86.0.4240.75 on a system with an Intel Core i7-9750H processor, an Nvidia GTX 1650 4GB GDDR5 Graphics Processing Unit, 1TB PCIe NVMe SSD, and a 60fps Full-HD monitor.

# Nodes	**# Edges**	**Frame rate**	**RAM used**	**Time to load**	**Time to load from .diva file**
400	1500	60 fps	10 MB	5 sec	1 sec
1k	3800	60 fps	13 MB	6 sec	1 sec
1.5k	5100	60 fps	14 MB	6 sec	1 sec
10k	100k	60 fps	38 MB	$\approx$ 25 sec	6 sec
5k	150k	60 fps	73 MB	$\approx$ 23 sec	8 sec
25k	700k	60 fps	159 MB	$\approx$ 1.7 min	$\approx$ 35 sec
35k	1.33M	60 fps	538 MB	$\approx$ 3 min	$\approx$ 1.1 min
68k	**4.5M**	**35 fps**	**925 MB**	$\approx$ **6 min**	$\approx$ **2.3 min**

One observes from Table 3.20 that under similar network configurations, `Epinet` and `NDlib-Viz` become unresponsive once the number of input nodes exceeds $10k$ and $6k$, respectively. Initially, the slightly higher time for `DiVA` to load the network can be attributed to the Force-Atlas algorithm (Jacomy *et al.*, 2014) that generates the layout before the network gets displayed. At $3k$ nodes (the largest network size available to compare all three systems), we observe that `DiVA` produces a small gain of 2.88 and 3.51 seconds over `NDlib-Viz` and `Epinet`, respectively. The difference in the performance of the three systems can, in part, be attributed to how the underlying packages handle diffusion simulations. Since both `NDlib-Viz` and `DiVA` rely on NetworkX and NDlib, we believe the performance boost in `DiVA` is due to (a) the superior graphing library in three.js and (b) the way we stream response from the backend to the frontend. It also makes `DiVA`'s visualisations relatively interactive and allows a high frame rate on large graphs. Further, the scalability of `DiVA` in terms of space and time is highlighted in Table 3.21. An edge set of $4.5M$ takes up less than $1kB$ of RAM. Our system can load and operate on networks ranging from as small as 400 nodes and $1.5k$ edges to networks as large as $68k$ nodes and $4.5M$ edges. The latter takes ≈ 6 minutes to load. If this same network is saved as `.diva` and loaded again, the response time reduces to 2.3 mins.

3.5.5 Limitations and future work

- **Tied to specific backend libraries.** The current version of `DiVA` uses NDlib and NetworkX for network modelling. Future work can extend the backend to other network libraries. NDlib has limited support for snapshot and interaction diffusion executions. We hope to leverage this setup to introduce support for dynamic and advanced networks in the future.
- **Limited support for node metadata.** The current visualisation and rendering employ nodes as numeric IDs. This limits support for metadata information to be displayed on the graph panel. We hope to incorporate more tools and feature sets, such as introducing spatial information, geotagging, user influence analysis, and deeper community analysis into `DiVA`.

3.6 Chapter conclusion

When reporting a tweet, X asks for other similar tweets on the offender's timeline that appear offensive to the reporter. More often than not, the motivation for an offender to engage with hateful content is a combination of internal and external influences. Thus, human social media interactions point to contextual nudging, either by elements on the platform or externally. This chapter provides a mechanism for curating, analysing, and modelling contextual signals. We highlight successful use cases in terms of reposting behaviour (`RETINA`) and the classification of hateful content (`HEN-mBERT`), both of which are improved by the infusion of endogenous and exogenous context. Note, in both setups, the accompanying datasets `ConInHate` and `GOTHate` are not touted as superior benchmark datasets since such a concept is complicated to define for the hate speech literature. Instead, we obtain a more 'generic view' of social media discourse. From our thesis so far, it is evident that employing richer textual embedding (either via better modelling or contextual signal infusion) helps reduce the information gap between computational models and human moderators, especially in the absence of observable intent. However, practitioners need to keep in mind that it is not just the inclusion of signals but the manner of infusion that is equally important. Setups such as in Figure 3.20 serve as an illustration for reducing the information gap between NLP and human decision-making, thereby reducing their burden.

> The superior performances of `ConInHate` with `RETINA` as well as `GOTHate` with `HEN-mBERT` reinstates our quantification of contextual information. The latter also corroborates the role of PLMs in modelling hateful signals. However, no single piece of auxiliary information is fully comprehensive on its own, and it is only a combination of multiple contextual signals that helps sufficiently differentiate the hateful users and posts.

Current and future state of data curation from OSNs. Through extensive data curation, analysis, and modelling, we have empirically established the role of unlabelled contextual signals in both endogenous and exogenous forms. However, this research relied on free and academic API access on X, additionally supported by 3rd party access. At the time of completing this thesis, both X and Facebook have turned off their free API access. These issues have caused a setback for the research community. Analogous to soliciting in the physical world but digitally, in the future, we can pay people to

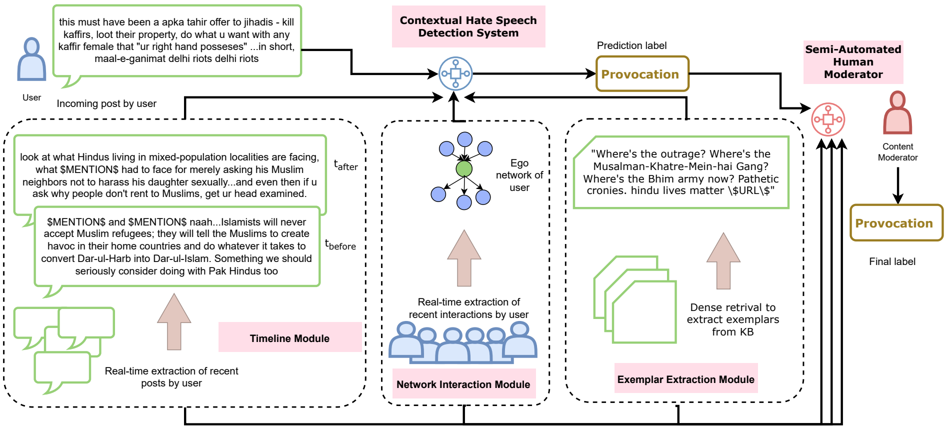


Figure 3.20: The desired pipeline for semi-automated flagging of hateful content. For an incoming post, we query the user's timeline and interactions to extract their most recent footprint. Additionally, we query the existing exemplar base. A combination of incoming posts and auxiliary signals is employed to detect contextual hate speech. Meanwhile, these signals and predicted labels are provided to content moderators to confirm the label.

share data about the groups they are part of and the content in those platforms they post and interact with (Chauchard and Garimella, 2022; Garimella and Chauchard, 2024). Another is to employ web archives, and instead of looking at OSNs, we perform narrative assessments of news and blogs as the source of spreading harm and propaganda leading to hate (Antoniak *et al.*, 2024; Yu *et al.*, 2021).

The last resort is to enrich existing textual datasets in hate speech literature with additional annotations, either manually (ElSherief *et al.*, 2021) or LLM-supported. These attribute-rich data sets can then be employed to study the contextual modelling of hateful signals. In the upcoming chapters, we explore this setup as a way to overcome the current challenges in the API constraints of OSNs.

CHAPTER 4

Uncovering Implicit Toxicity

"Behind the corpse in the reservoir, behind the ghost on the links, behind the lady who dances and the man who madly drinks, under the look of fatigue, the attack of migraine and the sigh, there is always another story, there is more than meets the eye."
- W. H. Auden; At Last the Secret is Out

4.1 Chapter introduction

Coded language and symbolism have shaped our world, from wars to dissents to art. Implicit hate speech is a form of coded language that has gained prominence on the Web, as it enables users to circumvent content moderation systems. A significant drawback of our datasets (`ConInHate` and `GOTHate`) curated so far is their lack of clear differentiation between implicit and explicit forms of hate. As implicit hate can manifest in various forms (Kruk *et al.*, 2024; Sap *et al.*, 2020; ElSherief *et al.*, 2021; Breitfeller *et al.*, 2019), to introduce the notion of implicitness in this thesis, we look into political attacks. Unlike hateful content, which is hard to curate (Kulkarni *et al.*, 2023), political parties often engage in verbal swordplay and politically charged insults, which worsen during elections. Here, we curate `PolAt` (**Pol**itical **At**tack), a code-mixed dataset from the Indian Assembly Elections of February 2022. Our analysis of `PolAt` reveals the power dynamics of incumbent vs opposition and how offline events trigger the increase in political attacks in online spaces. It also highlights the performance gap of NLP models when detecting implicit hate in the absence of contextual signals. From `PolAt` we observe that the skewness in the number of implicit samples contributes to its relatively lower performance. Moreover, based on our probing of contextual signals, we postulate that the information required to discriminate implicit hate would differ from the signals we have explored in the last Chapter.

To reduce the knowledge gap between implied and surface forms of implicit hate, we thus propose `FiADD` (**F**ocused **I**nferential **A**daptive **D**ensity **D**iscrimination). By employing an implicit-aligned context signal, `FiADD` brings the surface form of implicit content closer to its implied/intended form while increasing the inter-cluster distance among various class labels. We test the generalisability of `FiADD` on four use cases, detecting implicit hate, sarcasm, irony, and stance, in which surface and implied forms differ, and observe similar performance improvements. While the performance gap between explicit and implicit persists, the impact of `FiADD` on latent space clearly highlights the advantage of employing distance and contextual finetuning for implicit hate detection. In the last section of this chapter, we reflect on the tradeoff between standard performance metrics (relied heavily upon in this and the previous chapter) and bias evaluation metrics in the hate speech literature. Due to the nascent and multi-faceted nature of the work, the literature is chaotic in its terminology, techniques, and findings. As a first step, we put together a systematic study of the limitations and challenges of existing methods for mitigating bias in toxicity detection.

4.2 Political attacks and mud slandering in India

4.2.1 Motivation

Research gap. Any hateful attack focused on identity (like religion, caste, or gender) rather than political ideology should be discouraged (Chan *et al.*, 2021; Waseem *et al.*, 2017). Barring that, in a vibrant democracy, political attacks and criticisms are the norm (Petkevic and Nai, 2022). Despite the rich prevalence of direct attacks, most of them walk the thin line of provocation. Moreover, come election season, all media outlets, including the Internet, are buzzing with politically charged content. Both of these aspects provide an ideal ground to examine the nature of implicit attacks. Firstly, the significant gap in this analysis is the lack of Hindi toxicity datasets with implicit markers. Secondly, among the datasets on political examination that do exist, they are curated retrospectively once the whole event has passed (Jafri *et al.*, 2023). However, elections are a fast-moving landscape, and datasets studying them need to be curated more proactively.

Research questions. Having curated `PolAt` on a weekly basis from Twitter, we attempt to answer the following RQs:

RQ1: Based on manual annotations of implicit attacks, what can be said about the nature of attacks among the major political parties?
RQ2: How hard is it to perform computationally aided annotations for implicit attacks?
RQ3: How do the online and offline political events influence each other?

Contribution summary. We begin curating `PolAt` as a collection of tweets obtained on a biweekly basis from January to March 2022. This captures a month before and after the February 2022 Assembly elections. A political attack is treated as a subcategory of offence, non-overlapping with identity-based attacks such as hate speech (Solovev and Pröllochs, 2022; Schmidt and Wiegand, 2017). We manually quantify the power dynamics of self-promotion (Jakesch *et al.*, 2021) and negation pinned on the ruling party. Consequently, to aid large-scale analysis, we obtain pseudo-labels for the rest of the dataset via training a political attack detector for the Indic setting. Through `PolAt` we observe that the trends of engagements established via manual annotations hold at scale (Masud and Chakraborty, 2023). Finally, we showcase that during elections, noteworthy events in the physical world get magnified online (Nellis, 2023).

4.2.2 Background on Indian politics

India is the largest democracy and the fifth-largest economy in the world. In India, the legislative power is divided between the Union/central government and the state government. The former is led by the prime minister (PM) of the country, and the latter by the chief ministers (CMs) of the respective states. Direct elections are held for both legislatures every 5 years. The central election is also known as the general election. Meanwhile, the state-level elections are known as assembly elections. The current study examines the assembly elections of February 2022, held in five states of India – Uttar Pradesh (UP), Punjab, Goa, Uttarakhand, and Manipur. Before the elections, the Bhartiya Janta Party (BJP) was the majority in Goa, UP, Uttarakhand, and alliance-based power in Manipur. They retain power in four states, winning by majority in Goa, Manipur, and

Uttarakhand and by alliance in UP. Meanwhile, in Punjab, the power shifts from the Indian National Congress (INC) to the Aam Aadmi Party (AAP).

Table 4.1: The volume of posts in `PolAt` from the 17 political groups. We summarise the number of tweets (T), the number of unique users (U), the ratio of tweets to users (T/U), total retweets (R), average retweets per tweet (R/T), total likes (L), and average likes per tweet (L/T). The abbreviations for the political party are – Aam Aadmi Party (AAP), All India Majlis-E-Ittehadul Muslimeen (AIMIM), Apna Dal (AD), Azad Samaj Party (ASP), Bhartiya Janta Party (BJP), Bahujan Samaj Party (BSP), Bhagwa Kranti Sena (BKS), Goa Forward Party (GFP), Indian National Congress (INC), Jammu and Kashmir Peoples Democratic Party (PDP), Lok Janshakti Party (LJP), Rashtriya Janta Dal (RJD), Rashtriya Lok Dal (RLD), Samajwadi Party (SP), Sanyukt Samaj Morcha Party (SSM), Shiromani Akali Dal (SAD) and Shiv Sena (SHS).

Name	**# T**	**# U**	**T/U**	**# R**	**R/T**	**# L**	**L/T**
AAP	3537	14	252	1447949	409	6894606	1949
AIMIM	2179	10	217	650658	298	2984679	1369
AD	214	1	214	24525	114	148145	692
ASP	140	1	140	232700	1662	969690	6926
BJP	26879	37	726	14296955	531	72566346	2699
BSP	132	3	44	197609	1497	933682	7073
BKS	720	1	720	446423	620	2426117	3369
GFP	125	1	125	2269	18	7562	60
INC	12357	33	374	5617578	454	21960713	1777
PDP	59	1	59	6819	115	39289	665
LJP	90	1	90	484	5	3411	37
RJD	86	2	43	73987	860	517034	6012
RLD	164	1	164	18320	111	120412	734
SP	6480	10	648	3551946	548	18232421	2813
SSM	12	1	12	215	17	1064	88
SAD	514	3	171	33244	64	91111	177
SHS	124	2	62	92942	749	834949	6733

4.2.3 Dataset curation

Within the abundance of social media platforms, we once again focus on Twitter, given its popularity in Indian politics (Jafri *et al.*, 2023) and the ease of collecting data for academic research (as of early 2022). We configure the APIs to filter tweets based on geolocation to focus on India, but do not filter for language. By concentrating on 111 political leaders representing 17 political groups and one set of independent candidates, we amass the tweets biweekly from January 1 to March 31, 2022. The wide variety of leaders and parties provides variability in the political voices. Our dataset, `PolAt`, includes political leaders from – Aam Aadmi Party (AAP), All India Majlis-E-Ittehadul Muslimeen (AIMIM), Apna Dal (AD), Azad Samaj Party (ASP), Bhartiya Janta Party (BJP), Bhagwa Kranti Sena (BKS), Bahujan Samaj Party (BSP), Goa Forward Party (GFP), Indian National Congress (INC), Jammu and Kashmir Peoples Democratic Party (PDP), Lok Janshakti Party (LJP), Rashtriya Janta Dal (RJD), Rashtriya Lok Dal (RLD), Samajwadi Party (SP), Sanyukt Samaj Morcha Party (SSM), Shiromani Akali Dal (SAD) and Shiv Sena (SHS). We also examine the official Twitter handles of six political factions – BJP, INC, AAP, SP, BSP, and AIMIM. These six are the largest and most influential in UP and Punjab. We map different politicians to their parent party based on a politician's Twitter profile or through self-declaration of the most recent tweets. In ambiguous cases, we also check their Wikipedia page (if any) and official party bulletins. We term a political group as a *political party* when it officially contests in regional or national elections. On the other hand, we reserve the term *political syndicate* for a socio-religious organisation that does not contest elections, yet can be ideologically associated with various political parties. Bhagwa Kranti Sena (BKS) is the only syndicate in our dataset. Note that for our current analyses of political attacks, we do not differentiate between a syndicate and a party. We treat all 'political groups' capable of initiating attacks. As we observe a minimal set of independent candidates, we do not cover them in this discussion.

Overall, we curate $46k$ posts consisting of $14,450$ non-duplicate tweets from the official party handle and $32,017$ posts from politicians. To capture the diversity in interactions garnered by these parties, we look into party-wise user and tweet contributions along with average likes/retweets per tweet. Table 4.1 provides an overview of the same. Three out of five election states are in North India, where Hindi is spoken widely. Consequently, Hindi, English, and Punjabi are the most frequent languages in `PolAt`. While piggybacking on the timeline of UP and Punjab elections, Goa and Manipur also receive some attention. Unsurprisingly, but sadly, we also observe that Meitei (the official Manipuri language) does not even show up among the top 6 languages in our datasets. On the other hand, for Goa, we observe only one local party, the Goa Forward Party (GFP), with only $1.2k$ tweets and $7.5k$ likes.

4.2.4 Manual annotations

Definitions. Within the larger purview of offensive content, *political attacks can be defined as defamation and accusatory remarks made to project the opposing political groups negatively* (García Benítez-D'Ávila, 2022). It should be noted that while these political attacks fall under the more extensive umbrella of derogatory content, they do not qualify as hate speech because politicians and political parties are not vulnerable entities (Zahrah *et al.*, 2022; Masud *et al.*, 2021). It qualifies as hate only if an attack is on personal identities (race, religion, caste, etc.). Hence, in this work, we stick to the term political attack. A *direct/explicit* attack targets political affiliations, ideologies, and policies and may employ political name-calling and slurs (about politics). Meanwhile, an *indirect/implicit* attack is expressed with taunt, sarcasm, irony, etc., and may require cultural/contemporary referencing for comprehension. For example, consider the following post from AAP commenting on how, unlike BJP and INC, which only honour the respective party affiliates, AAP chooses to celebrate only freedom fighters like Bhagat Singh. Thus, 'implicitly attacking' other parties.

#1 "BJP: Savarkar और Hedgewar की तस्वीर क्यों नहीं लगाई? Congress: Indira Gandhi, Rajeev Gandhi, Sonia Gandhi की तस्वीर क्यों नहीं लगाई? मैं कहता हू कि इन सबकी तस्वीर आप लोग लगा लो, हम तो बाबा साहेब Ambedkar और Bhagat Singh जी की ही तस्वीर लगाएंगे।" *(BJP puts up Savarkar and Hedgewar's photos, Congress puts up Indira Gandhi, Rajeev Gandhi, Sonia Gandhi's photos. I say you pay tribute to these photos if it pleases you, but we will only put up Bhagat Singh's photos.)*

Table 4.2: Mapping annotations from first phase (L1) to second (L2). For L1, 'N, I, E' stand for the neutral, implicit attack, or explicit. Meanwhile, for L2, 'P, D, B, N' stands for promotion, demotion, both, or neutral.

L1/L2	P	D	B	N
N	415	13	18	249
I	15	219	92	3
E	6	517	172	1

Annotator details. Two Indian female volunteers aged 22 and 26 help with the annotations. They are proficient in English and Hindi. They are knowledgeable about Indian politics and social media. A third annotator is involved in case bases to resolve disagreements in annotations. The annotators employ Google Translate when translating from local languages. None of the annotators are associated with any political organisation.

First-level annotations. In this phase, firstly, the annotators separately label a random sample of 100 posts each, refining the definition described above until an IAA of 0.70 (Cohen's κ) is achieved. Following this, in the second round, each annotator contributes ≈ 750 annotations. In the end, $1.7k$ posts are labelled across 695 neutral, 696 explicit, and 329 implicit instances of political attacks.

Second-level annotations. In the second phase, for tweets by BJP, INC, AAP, and SP (politicians and party handles), the annotators come together and annotate 4 secondary labels. These annotations are only performed for the $1.7k$ manually annotated posts filtered for the party. The aim is to find whether the content indicates self-promotion/advertisement, an attack/denouncement of opposition, or both, or neither. Here, promotion is expressed in the form of any positive sentiment, promise, or accolade expressed by the party for itself. For example, SP uses the hashtag "बाइस में बाइसिकल" *(Cycle in* 2022*)* to refer to its election symbol of a cycle as an indicator of SP turning the wheels and coming to power. On similar lines, denouncement refers to any form of negative sentiment expressed by another party. In some cases, the parties tactfully promote themselves while demoting others. Going back to Example #1, in which AAP talks about the kind of leaders the BJP/INC pay homage to, they continue to state that the party will not pander to any political leader, old or new, and only pay tribute to freedom fighters, therefore, qualifying as 'both' – self-promotion and opposition's demotion.

While coarse-grained annotation captures the overall sentiment or explicitness, fine-grained annotation captures the intent, akin to aspect-based sentiment analysis. From Table 4.2, we see that samples marked as neutral are more likely to be either promotional or informational. Meanwhile, content marked as explicit is highly susceptible to demoting/denouncing in nature.

Note on annotation bias. While it is impossible to combat biases in data curation, adequate measures are required at each pipeline step. These combined measures help annotators perform as contextually aware annotations as possible. In our case, the annotators actively follow the latest election news and are able to refer to multiple sources for reference. Additionally, during the annotation process, when the attack is not apparent from the posts, the annotators also make use of the post's metadata.

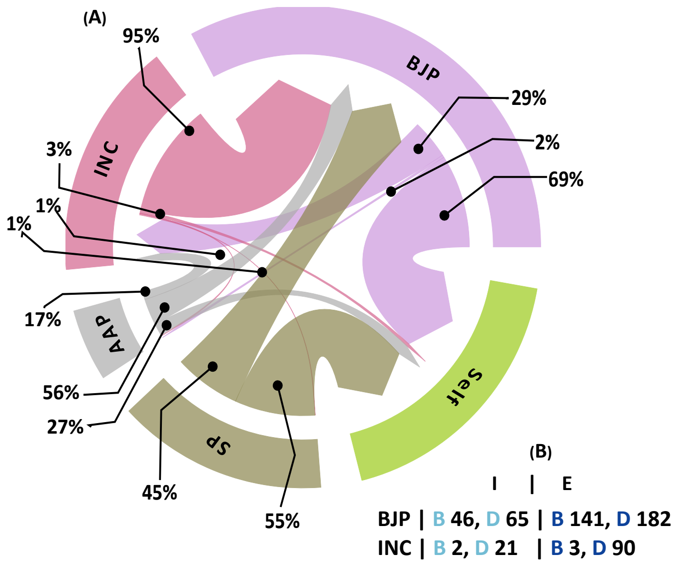


Figure 4.1: An overview of promotion and demotion by four parties – BJP, INC, AAP, and SP. (a) The weighted and directed chord diagram represents the dynamics of self-promotion and denunciation/attack on opponents among the four parties. The interactions are based on the manually annotated promotions and demotions (*aka* advertising vs attacking) of $1.7k$ tweets. Given that the parties under consideration are not allies, we can safely assume that comments directed at other parties are not promotional. Self-promotion is captured by edges directed at the self (green) chord. We also annotate each chord with the percentage breakdown of tweets involved in a particular action, e.g., $X \xrightarrow{p\%} Y$ represents the p% of the tweets by X that are directed towards Y. (b) A breakdown of denunciation by BJP and INC. Here, I and E represent the implicit and explicit attack labels; meanwhile, B and D capture whether the attack is either promotion and demotion or purely denouncing.

4.2.5 Extent of promotion and demotion

To answer the first research question, we manually examine the power dynamics between the ruling and opposing parties through the lens of promotion and criticism. BJP is the ruling party at the national level and in four out of five states covered in these elections (Harriss, 2015) (as of January 2022). They participate in the discourse from a position of security with 69% of their annotated samples being self-promotion as highlighted in Figure 4.1(a). One observes the same regarding hashtags and name-calling they employ. We observe that their hashtags for self-promotion heavily outnumber those demoting the opposition. While these hashtag frequencies are representative of the curated $46k$ post, the proportions should also scale in the wild. BJP employs self-promoting hashtags like “यूपीमांगेभाजपा” *(UPDemandsBJP)* and “#TripuraWelcomesModiJi” with frequencies of 120 and 35. While other hashtags like “#भगवामय उत्तराखंड” *(Uttrakhand will be painted orange)* and “#चप्पा चप्पा भाजपा” *(BJP at every nook and corner)* with frequencies of 39 and 75 have an undertone of intimidation, yet they still display dominance and self-promotion. Meanwhile, BJP rarely utilises explicitly-targeted hashtags such as “#अपराधी वाली सपा” *(SP's inherit Terrorism)* or “#KejriwalAgainstHindus”. Opposition parties with fewer resources need to be more dexterous about promoting themselves. They bank on the incumbent's lack of sympathy toward public issues to launch negative campaigning against the BJP. AAP and SP display this behaviour. They balance their resources for attacking the ruling party and self-promotion. The hashtags they employ reflect the same. While attacking the BJP, they adopt a combination of explicit and implicit hashtags such as "#BJPKeGunde" *(BJP's goons)*, "#DalitVirodhiModi" *(Anti Dalit Modi)*, and "#संविधानद्रोही_BJP" *(BJPisAntiConstitution)*. Note that we employ the presence of "#" at the start of a word to capture unique hashtags and their frequencies.

To further analyse the distribution of attacks and promotions among $1.7k$ manual annotations, we employ the second level of manual annotations for AAP, INC, SP, and BJP. With INC, we observe a ratio of $7:1$ for attacking vs. advertising. It is currently the largest opposition party. Therefore, INC will likely criticise the BJP at both national and regional levels. This behaviour is corroborated via Figure 4.1(a), in which we observe a higher density of attacks directed from INC to BJP. Despite being one of India's largest and oldest parties, the INC has recently seen a decline in voter share and political power. This could be one of the reasons why the BJP responds with a single attack for every three attacks initiated by the INC (Figure 4.1(a)). For reference, from 2017 to 2022, INC's vote share went down from 28.7% to 23.7% in Goa, 38.8% to 23.1% in Punjab, 6.3% to 2.4% in UP, and 35.3% to 16.9% in Uttarakhand. Their only success comes from Uttarakhand, where their share increased from 33.8% to 38.2%. Meanwhile, the volume of promotion and denouement by SP is more balanced. It again reflects the power dynamics of these parties in Indian politics. SP also opposes the BJP, but they are not as huge as the INC. Therefore, they invest their energy more equitably by focusing majorly on UP, using promotional jiggles like "#बाइसमेंबाइसिकल" (*2022 will be the year of bicycle)*, as well as intimidating BJP with hashtags like "#भाजपा खत्म" *(BJP is over)*. This behaviour is visible through equal arms of promotion and demotion originating from SP in Figure 4.1(a).

Note on the BJP-INC dynamics. The observation that the BJP attacks the INC less in proportion at first seems counterintuitive. However, one must interpret this observation exclusively in terms of the elections with a focus on UP. In UP's last assembly elections (2017), INC won only 7 seats while BJP won 312. Similar trends are observed in Goa, Manipur, and Uttarakhand. BJP has been able to hold its popularity until the next elections. It means that in 2022, they contest elections unassailably, represented by the 69% of their posts being about rallies, visits, or self-promotional information. In the rest of their denunciation of other parties, they still spend the majority on INC (29%). In absolute value, BJP still contributes significantly (compared to INC) to implicit (65 vs. 21) and explicit attacks (182 vs. 90) (Figure 4.1(b)). To have any significant impact on the ground, INC plays defence, which increases its attack against the BJP. BJP can denounce from a secure position, leading to a skew in the proportion of attacks by these parties, which could be one reason for the observed behaviour.

4.2.6 Large-scale annotations

Given the limited resources for manual annotation, it is desirable to have some form of computational assistance for large-scale labelling of attacks. Based on our experiment with NLP systems for classifying hate speech in the previous Chapter, we experiment with two setups:

- **N-gram based.** We extend the Davidson's Davidson *et al.* (2017) model. The original work employs Logistic Regression to detect hate, offence, and non-hate in English Tweets surrounding the USA. However, given Indic languages' cultural and semantic diversity, we add language-specific features like Indic stopwords and Indic tokenisation (Kakwani *et al.*, 2020). We combine the preprocessed n-gram features and the tweet's metadata as the final input to a Logistic Regression classifier.

- **PLM-based.** We employ mBERT (Devlin *et al.*, 2019), finetuned with a simple classification head.

Experimental setup. The TF-IDF and POS-tagger vectorisers are initialised for a maximum feature size of 150 and min/max$_{\text{df}}$ of $5/0.75$. We employ the n-gram setup with $n = 1, 2, 3$. The rest of the hyperparameters are default. For mBERT, we employ a batch size of 32, a maximum sequence length of 128, and set the AdamW optimiser with a learning rate of $3e-5$. In accordance with our previous observations with mBERT in the last chapter, we set the last six layers to be trainable. We begin with binary classification settings where the explicit and implicit labels collapse into a single class. We then extend our models for the 3-way classification with neutral, explicit, and implicit classes. Note that the binary classification is only performed for sanity testing. The final model is a 3-way classifier. We perform 3-fold cross-validation with class weighting and report the macro-F1 and ROC-AUC.

Performance metrics. The macro-F1 of the TF-IDF setup staggers at 0.80 in 2-way classification and drops to 0.60 under 3-way classification. Its performance on implicit class is abysmal at 0.44 macro-F1. On the other hand, the mBERT system achieves a macro-F1 of 0.98 in 2-way classification, which drops to 0.91 under $3-$ way classification. Here, even the implicit class enjoys a macro-F1 of 0.84. Further, under 2-way classification, the TF-IDF and mBERT models report a ROC-AUC score of 0.79 and 0.97, respectively. Meanwhile, under 3-way classification, the TF-IDF and mBERT models report an ROC-AUC score of 0.71 (0.71) and 0.95 (0.95), respectively, for the One-vs-One (One-vs-Rest) multiclass setting.

Labelling the attacks. The manually annotated $1.7k$ posts are henceforth referred to as *annotated samples* (Figure 4.2(a)). Given that the mBERT-based setup significantly outperforms TF-IDF, we employ the former for generating pseudo-labels for the rest of the samples in `PolAt`. This subset is known as *predicted samples* (Figure 4.2(d)). Note that the large-scale annotations are only performed for the first level of labels, leading to $23,838$ neutral, $17,771$ explicit, and $4,858$ implicit samples.

4.2.7 Analysis of attacks at scale

Employing the predicted labels, we can now examine whether significant differences occur in the proportion of attacks between the manual and predicted annotations.

Frequency of attacks. Since BJP and INC are India's two largest political parties, they are also the most active, making up for 75% of the tweets produced in `PolAt`. BJP and INC contribute around $23.8k$ and $11.8k$ tweets, respectively. Meanwhile, BKS and ASP display an intriguing social media presence. While BKS and ASP have a low volume of tweets, their posts receive considerable retweets and likes. Upon analysing their posts, we observe that BKS either posts in support of the BJP or directly attacks other parties. Tweets by ASP are all politically attacking the BJP. The percentage of attacking vs neutral posts within full datasets in BKS (48.3%) and ASP (60%) further corroborates this pattern. An example of BKS supporting BJP in a neutral manner, which

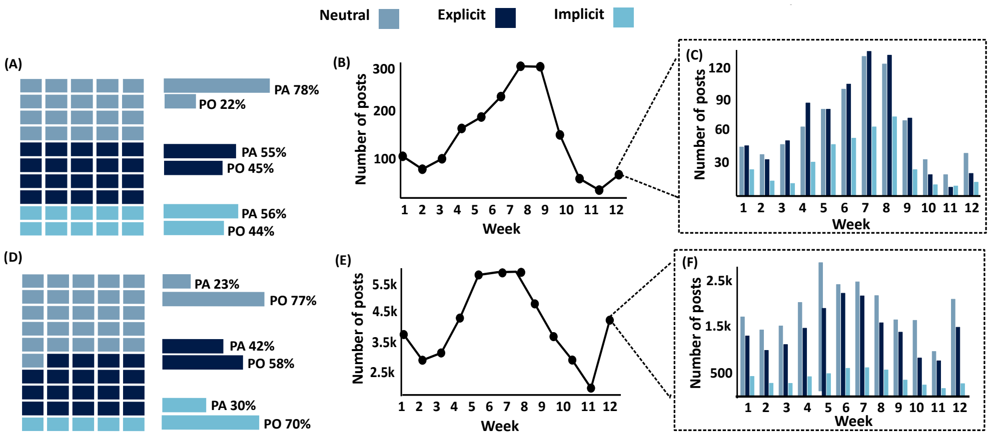


Figure 4.2: Overview of `PolAt`, with a breakdown of neutral, explicit, and implicit remarks. The top row highlights the pattern of attacks for the manually annotated samples; the bottom row highlights the same set of patterns for predicted samples. Each box in the pictograms presents 1% of the respective class for the manual (a) and predicted (d) labels. The two user groups in the curated dataset are Political Party handles (PY) and Politicians (PO). Sub-figures (b) and (e) capture the week-wise volume of posts from January to March based on our annotated and predicted samples, respectively. Meanwhile, (c) and (f) represent the granular weekly proportion of neutral, explicit, and implicit attacks obtained from our annotated and predicted samples. The x-axis represents the nth week, and the y-axis represents the volume of posts.

received $3,371$ and $1,911$ retweets and likes, is “पूरा यूपी डोल रहा है, योगइ योगइ बोल रहा है!” *(Entire UP is mesmerised and chanting Yogi Yogi!)*. A similarly high interaction post explicitly attacking the BJP coming from ASP is “General Dyer in the guise of a sage crossed all limits of cruelty. So ruthlessly that even the sticks, the British started to feel embarrassed. First, students were beaten fiercely; now, an FIR was lodged against 1000 students. The youth will answer Yogi Ji, who is preparing to criminalise the future. #RRBNTPC_Scam”. It receives $4,663$ $(15,156)$ retweets (likes).

Proportion of attacks. Among the annotated samples in Figure 4.2(a), the ratio of attacking to neutral posts is 3:2, which is anticipated given the election season. Interestingly, trends from the annotated samples translate to the predicted samples obtained via pseudo-labelling (Section 4.2.6). Among predicted samples, we observe a ratio of neutral vs. attacking being 1:1. Direct attacks in manual and predicted samples overshadow implicit ones by 2:1 and 3:1, respectively.

The week-wise spread of attacks peaks in February when the elections are held and dwindles after that. The trend is visible in both annotation sets.

When looking at the overall volume of posts in Figures 4.2(b) and 4.2(e) as well as the label-wise breakdown in Figures 4.2(c) and 4.2(f), February appears as the most active month. Interestingly, we find that the mean retweets/likes in explicit posts are significantly more (t-test) than in implicit posts for predicted samples, but not significant for manually annotated samples. For pseudo labels, the t-test for median retweets/likes yields a p-value of $1.030e^{-5}$/$1.164e^{-17}$ and an effect size of $0.654/1.366$. It points toward the fact that, in general, explicit posts are more likely to receive more engagement than their implicit counterparts, as the former is easier to handle by a larger audience (Masud *et al.*, 2022). Meanwhile, for manually annotated samples, the t-test for median retweets/likes yields a p-value of $0.79/0.51$ and effect size of $0.11/0.28$. We pin this difference in a pattern on the pseudo-labelled set's $\approx 20X$ larger size. When we randomly select $1.7k$ samples from the predicted set, the tests for retweets/likes mimic that of the manually annotated samples with a p-value of $0.684/0.006$ and effect size of $0.176/1.301$.

4.2.8 Online and offline influence

Looking beyond numbers and labels, elections have real-world implications. It is imperative to discuss how the online and offline events interact. Appendix D enlists some additional instances of name-calling.

A gamble for AAP. AAP's most successful campaign is in Punjab, winning 92 out of 117 seats. AAP tactfully curates a combination of promotional and denouncing hashtags (Siddarth *et al.*, 2021). On the one hand, they bank on the incumbency of the existing parties. Here, they employ positive and promotional hashtags like "#PunjabDiUmeedAAP" *(AAP is Punjab's hope)* and "#AAPKeGovtSchools" *(Government Schools by AAP)*, which beckon AAP for bringing a systematic change. On the other hand, the alleged involvement of the state's then chief minister in scams prompted the use of accusatory hashtags like "#DarrGayaChanni" *(Channi is afraid)* and "#RetaChorChanni" *(Sand Mafia Channi)*. Interestingly, a similar tactic by AAP of promoting the so-called *Delhi model of development* did not favour Goa. It is safe to say that the level of incumbency and public dissatisfaction is equally essential for voters, apart from political invectives.

Rising popularity of SP. The number of seats won by SP in UP went from 47 in the year 2017 to 111 in 2022. SP is also the third voluminous party in our curated dataset with an average retweet and like counts of 650 and 3365, respectively. While other parties could achieve the same numbers on social media, their user engagements did not translate into vote share. The rising popularity of SP in the online world can be both an antecedent and a consequence of its rising popularity on the ground.

Not all influence is positive. Political attacks can lead to internal/domestic extremist groups acting in ill faith (Piazza, 2020). During one of the campaigns, Punjab's then CM Charanjit Singh Channi made a statement– "Do not let UP, Bihar ke bhaiya enter Punjab" *(Do not allow men from UP and Bihar to enter Punjab)*. He later clarified that it was meant for the people of Punjab not to let political leaders like AAP from Delhi enter Punjab politics. However, his clarification comes too late. An implicit attack on AAP ends up as a barrage of explicit attacks against him and other female politicians who accompanied him (Rheault *et al.*, 2019). This incident alone spiked the volume of overall and explicit posts as captured in week 7 of Figures 4.2(e) and 4.2(f).

4.2.9 Limitations and future work

- **Lack of equity in coverage.** The overwhelming quantity of tweets and subsequent discussions hardly covers the states of Goa and Manipur. In the future, we ought to opt for more state-level political entities. Another way to increase representation is to employ tweets from local news and city or state-specific hashtags.

- **Role of latent influences on election outcomes.** Owing to API rate limits and our computing resources, the data we collect is only a subset of the overall electoral social media data. It does not account for political activists and citizens, as well as news organisations. As multiple online and offline events can influence election outcomes, in this study, we perform only anecdotal analysis and refrain from making win/loss predictions. In the future, combining social media data with exit polls and in-person interviews can help develop outcome prediction models.
- **Gaps in model-based labelling.** We recognise, as a part of manual annotations, that performing promotion and demotion classification requires knowledge of Indian political parties and the relationships among them. As these secondary labels are recognised based on span/attribute level indicators, they are difficult to replicate at scale. Meanwhile, for the first-level annotations, PLMs outperform classical methods in implicitness detection. However, there remains a gap in the overall (0.91) vs implicit class (0.84) macro-F1. In future work, techniques that augment the world knowledge of the PLMs should be explored.
- **Contextual signals for Political Attacks.** Based on manual assessment of political statements from Twitter, one apparent contextual signal is the use of country (in this case, India) specific abuse slurs. One other way to expand the coverage of slurs is longitudinal analysis of dog whistles and metaphors (Mendelsohn *et al.*, 2023; Mendelsohn and Budak, 2025). Another easy-to-use signal can be an emotion label via multilingual LLMs.

4.3 Focal inferential infusion for implicit hate detection

4.3.1 Motivation

Research gap. From our last research, we manually observed that on the surface, implicit hate appears lexically and semantically closer to neutral statements. However, we need to establish empirically, “to what extent do the implicit and neutral spaces overlap in latent space?" Moreover, our secondary-level annotations of `PolAt` reinstate that inferring the underlying stereotype and implied hatefulness in an implicit statement requires a combination of multi-hop reasoning with sufficient cultural reference and world knowledge. To overcome this information gap, studies have mainly explored the infusion of the external context in the form of knowledge entities, either via KG tuples (ElSherief *et al.*, 2021) or Wikipedia summaries (Lin, 2022). A recent examination of the quality of knowledge infusion for implicit hate reveals that KG tuples fail to enlist information that directly connects with the implicit entities, acting more as noise than information (Yadav *et al.*, 2024). Borrowing from our experience with contextual signals in the last chapter, we postulate that for implicit hate, explanatory signals should be better suited than numerical ones. While exogenous signals like news articles do provide a breadth of context, they still fail to provide a 1-1 mapping. Determining which news article or stanza within an article actually leads to implicitness will require making the incoming post less implicit (a chicken-and-egg problem). There is a gap in 1-1 mapped contextual signals that can improve implicit hate detection.

Research questions. Based on existing implicit hate classification datasets, we attempt to answer the following RQs:

RQ1: How separated are the latent spaces of implicit, explicit and neutral statements?
RQ2: Can the cluster separation be improved by infusing better context during PLM finetuning for implicit hate detection?

Contribution summary. To this end, we first establish the closeness of implicitly hateful samples from non-hateful ones and use it to motivate our model design (Figure 4.3). We then examine the distance-metric-based Magnet Loss (ADD) (Rippel *et al.*, 2016), which is frequently employed in computer vision, and adopt it for the NLP setting. We propose **F**ocused **I**nferential **A**daptive **D**ensity **D**iscrimination (`FiADD`) as a pluggable unit in the PLM finetuning pipeline for the task of hate speech detection as well as other implicit text-based tasks. Here, the distance metric is combined with 1-1 mapped free text sentences explaining the underlying implicit hate to make the context more 'explicit' in nature. While we touched upon the theme of implicitness via political attacks in `PolAt` at the time of curating this thesis, Hindi hate speech datasets with markers for implicitness did not exist. Therefore, for our proof of concept, we restrict this study to English-only datasets. Apart from employing the `LatentHatred` dataset to support our experimental setup, we manually generate implied explanations/descriptions for 798 and 404 English implicit hate samples for `AbuseEval` and `ImpGab`, respectively. Our exhaustive experiments, analyses, and ablations highlight how the latent space evolves under `FiADD` when trained for the task of implicit hate detection (Masud *et al.*, 2024*a*).

4.3.2 Dataset curation

Among the myriad datasets on hate speech (Vidgen and Derczynski, 2021; Poletto *et al.*, 2021) in English, only a few have annotations for "implicit" hate. From among the hate speech database[1] (Vidgen and Derczynski, 2021) and ACL anthology[2] (we look at the results from the first two pages out of 10), we search with the keyword 'implicit' as an indicator of whether the label set contains 'implicit' labels. Filtering for modality, language, and human-generated samples, we shortlist `LatentHatred` (ElSherief *et al.*, 2021), `AbuseEval` (Caselli *et al.*, 2020), and `ImpGab` (Kennedy *et al.*, 2022). They are all English-based text-only datasets curated from the web with explicit, implicit, and non-hate labels. For `LatentHatred`, we employ the first level of labels and the existing manual annotations of implied hatred (a free-text short explanation of the underlying/implied/intended hatred) for implicit samples. Meanwhile, `AbuseEval` and `ImpGab` do not have the implied counterparts. Hence, we manually annotate the implicit hate samples of these datasets with their implied meaning generated as free text, as outlined in Table 4.4.

Table 4.3 provides a label-wise distribution of the three datasets. Note, at this point, we discard using the implicit hate datasets SBIC and Toxigen, as both of them only have implicit samples and are not suitable for 2-way or 3-way hate classification.

4.3.3 Dataset annotation

Implied contexts are succinct statements that make explicit the underlying stereotype. Note that the implied context cannot be considered a comprehensive explanation for

[1] https://hatespeechdata.com/
[2] https://aclanthology.org/

Table 4.3: Hate datasets employed in evaluating FiADD

Dataset	Source	Labels		
		N-OFF	EXP	IMP
AbuseEval	Twitter	11173	2129	798
ImpGab	Gab	25102	2159	404
LatentHatred	Twitter	13291	1089	7100

Table 4.4: Some sample posts from AbuseEval and ImpGab along with their implied annotations. We also provide the cross-annotator scores and remarks.

Dataset	Implicit Post	Implied Annotation	Score	Remarks
AbuseEval	@USER How dare you tell me what Antifa is doing is right. They are not fighting fascism. They are fighting for their own stupid cause. They are further from fixing fascism than the Republicans themselves,	Anti-fascists are stupid and good for nothing.	5	Captured explicit target entity and the broader abuse in context
AbuseEval	@USER And this traitor thinks he is?	He thinks highly of himself.	2	Captured explicit target entity but fails to note broader abuse (traitor or anti-national) in context
ImpGab	I said it, and I stand by it! Blacks get almost all of their cultural cues and information from rap and sports. It's not like they're reading books or anything. URL	Black people don't read books. Black people are less intellectual	5	Target is identified correctly. An alternate implication can be that black people are illiterate.
ImpGab	You missed out on his post saying there are still many Jews on #Gab! And didn't see his followers cuss out kike?! Good that you didn't see he claims himself a "racist capitalist"! I'm glad you missed ALL of them so you can continue to support him. AND excuse me, but he doesn't have much intelligence based on some of his posts. I'm not kike & not stand w/ any sides!	Christian are anti-Jews	3	Could be referring to a specific right-winged politician. Alternative meaning can be: You are as racist and anti-semitic as the leaders you follow.

implicit hate, but rather a more explicit understanding of the underlying subtle connotations.

Annotator details. Two expert annotators (*A and B*), one male and one female social media expert aged 29 and 35, volunteer to annotate the free text explanations.

Annotation framework. For AbuseEval and ImpGab, perform the annotations based on the following guidelines on tabular data:

- Implied meaning should consider the post's author's perspective (within the post's context).
- Implied meaning should emphasise the post's content only.
- Annotations must be explicitly associated with the target entity.
- Annotations must contain a broader abusive context for the given post.
- Annotations should balance lexical diversity and uniformity with respect to abuse towards a target group.

Annotation agreement. For our use case, annotation agreement scores help establish how well-aligned and coherent the explicit connotations are. To carry out the assessment, annotators A and B exchange a random sample of 30 annotation pairs. They score the pairs on a 5-point Likert scale (Likert, 1932), with 5 being the highest agreement. For AbuseEval, we obtain a mean agreement of 4.13 ± 1.13, and 4.07 ± 1.41 for ImpGab. Table 4.4 lists some sample annotations and their agreement scores. Further, a third expert (a 24-year-old male) conducts an independent survey using the above metric on the other set of random 30 samples. As per annotator C, for AbuseEval, we obtain a mean agreement of 4.55 ± 1.09 and 4.41 ± 1.15 for ImpGab. This independent assessment corroborates the annotation process, as annotator C did not participate in the initial annotations but observed similar alignment scores.

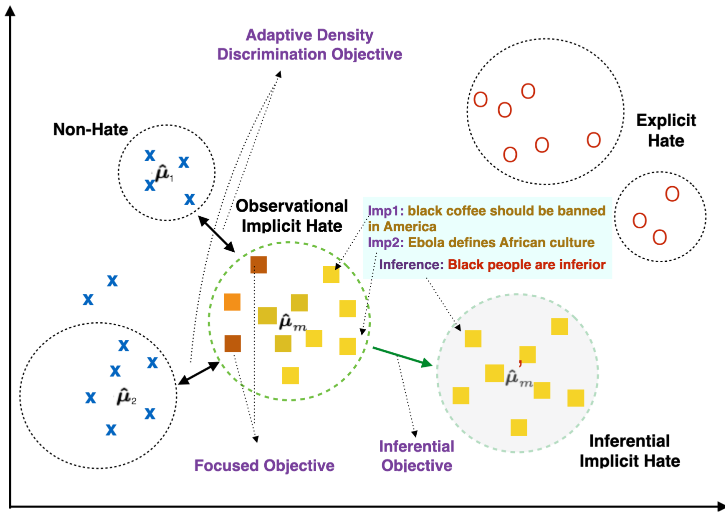


Figure 4.3: Intuition behind `FiADD` as applied to implicit hate detection, combing three objectives: (i) adaptive density discrimination to discriminate local clusters, (ii) focus on penalising samples near the discriminant boundary (*a.k.a* non-uniform weightage to cluster samples, and (iii) reduction in the latent-space difference between the surface and semantic form of the implicit hate.

4.3.4 Cluster separation of implicit hate datasets

Why is implicit hate hard to detect? Figure 4.3 shows two statements – "*Imp1: Black coffee should be banned in America.*" and "*Imp2: Ebola defines African Culture.*" Both Imp1 and Imp2 are statements that (i) do not contain any abusive or slur term, (ii) do not directly mention any target entity, but (iii) make indirect references to the target entity. These statements may not appear harmful unless the implied context is provided, i.e., "*Black race is inferior.*" In light of this implied context, the non-hateful surface form now expresses explicit hate. However, for the moderator (human or machine) to reach this inference, they must have sufficient cultural context or implied knowledge.

Clustering setup. For three implicit hate speech datasets, i.e., `LatentHatred` `ImpGab` and `AbuseEval`, we embed all the samples for a dataset in the latent space using a 768 dimensional CLS embedding from BERT. The embeddings are not finetuned on any dataset or task related to hate speech, so as to reduce the impact of confounding variables. We then consider three clusters directly adopted from the implicit, explicit, and non-hate classes. We also record the pairwise average linkage distance (ALD) and average centroid linkage distance (ACLD) among these clusters.

As the name suggests, for ACLD, we first obtain the embedding for the centre of each cluster as a central tendency (mean or median) of all its representative samples and then compute the distance between the centres. This distance indicates the overall closeness of the two centres, which, in our case, measures the extent of similarity between the two classes. We also assess the latent space more granularly via ALD. In ALD, the distance between two clusters is obtained as the average distance between all possible pairs of samples where each element of the pair comes from a distinct group. It allows for a more fine-grained evaluation of the latent space, as not all data points are equidistant from each other or their respective centres.

Formally, consider a d-dimensional set of data points $e_i \in E$ where each point (e_i) belongs to one of the N clusters c^n. For such a system $\mu^n = \frac{1}{|c^n|}\sum_{e_i \in c^n} e_i$ is the

Table 4.5: The L1-norm between neutral (N) and explicit hate (E), as well as neutral and implicit hate (I) samples based on ALD and ACLD.

Dataset	**ALD**		**ACLD**	
	N-E	**N-I**	**N-E**	**N-I**
`AbuseEval`	132.73	128.81	96.69	94.94
`ImpGab`	132.32	131.62	97.62	97.08
`LatentHatred`	125.12	122.59	90.86	90.69

cluster center. For any two clusters a and b, $ACLD^{a,b} = dist(\mu^a, \mu^b)$. Meanwhile, $ALD^{a,b} = \frac{1}{|c^a|*|c^b|} \sum_{e_i \in c^a \& e_j \in c^b} dist(e^i, e^j)$.

The intuition behind using both ACLD and ALD stems from the fact that online hate speech is part of the larger discourse on the Web. Thus, it is possible that at the level of individual data points, labelling an isolated instance as hateful is hard. Furthermore, some implicit samples may be closer to the explicit hate samples in terms of lexicon or semantics. On the other hand, it is also possible for some non-hate samples to contain slurs that are commonplace and context-specific but not objectionable within the community (Diaz *et al.*, 2022; Röttger *et al.*, 2021). ACLD and ALD allow us to capture these dynamics at macroscopic and microscopic levels.

Observations and takeaways. From Table 4.5, we observe that under both ALD and ACLD, non-hate is closer to implicit samples. As expected, ALD shows more variability than ACLD. It follows from the fact that the mere presence of a keyword/lexicon does not render a sample as hateful. Similarly, the use of sentiment labelling will be restrictive given the difference in surface-level neutral yet implied negative inclination of implicit hate speech.

Stemming from these observations, we see a clear advantage of employing a distance-metric approach that can exploit the granular variability in the latent space. The proposed model, as motivated by our empirical observations, is outlined in Figure 4.3.

> `Adaptive Density Discrimination` (ADD) based clustering loss, which optimises the inter and intra-clustering around the local neighbourhood, directly maps to our problem of regional variability among the hateful and non-hateful samples. Further, our observations motivate the penalisation of samples closer to the boundary, which is responsible for increasing variability.

4.3.5 Background on adaptive density discrimination (ADD)

Here, we briefly outline `Adaptive Density Discrimination` (ADD) (Rippel *et al.*, 2016), which forms the backbone of our proposed framework. ADD is a clustering-based distance metric. It evaluates the local neighbourhood or clusters among the samples after each training iteration. At each epoch, after the training samples have been encoded into vector space, ADD clusters all data points within a class into K representative local groups via K-means clustering. The subclusters within a class help capture the inter/intra-label similarity in the local neighbourhood. If there are N classes, then each training sample will belong to one of the $N * K$ subclusters.

Given that mapping and tracking distances among all N*K groups is computationally expensive, ADD randomly selects a reference/seed cluster I_s^c representing class C and then picks M imposter clusters from local neighbourhood $I^{c'}_{s1}, \ldots, I^{c'}_{sm}$ but from disparate classes ($c \neq c'$) based on their proximity to seed cluster. To understand the concept of seed and imposter clusters better, consider the three-way hate speech classification task with implicit, explicit, and non-hate labels. As we aim to distinguish implicit hate speech better, we select one of the implicit hate subclusters as the seed. Consequently, the impostor clusters will be from explicit hate or non-hate classes. ADD then samples D points uniformly at random from each sampled cluster. For the dth data point in m^{th} cluster, r^m_d is its encoded vector representation, with $C(.)$ representing the class for the sample under consideration. Subsequently, $\mu^m = \frac{1}{D}\sum_{d=1}^{D} r^m_d$ acts the mean representation of mth cluster. Here, ADD applies Equation 4.1 to discriminate the local distribution around a point:

$$p^{ADD}(r^m_d) = \frac{e^{-\frac{1}{2\sigma^2}\left\|r^m_d - \mu^m\right\|_2^2 - \alpha}}{\sum_{\mu^o : C(\mu^o) \neq C(r^d_m)} e^{-\frac{1}{2\sigma^2}\left\|r^m_d - \mu^o\right\|_2^2}} \tag{4.1}$$

Here, α is a scalar margin for the cluster separation gap. The variance of all samples away from their respective centres is approximated via $\sigma^2 = \frac{1}{MD-1}\sum_{m=1}^{M}\sum_{d=1}^{D}\left\|r^m_d - \mu^m\right\|_2^2$.

After each iteration, as the embedding space gets updated, so do the subclusters; this lends to a dynamic nature to ADD. The overall loss is computed via Equation 4.2.

$$\ell(\Theta) = \frac{1}{MD}\sum_{m=1}^{M}\sum_{d=1}^{D} -\log p^{ADD}(r^m_d) \tag{4.2}$$

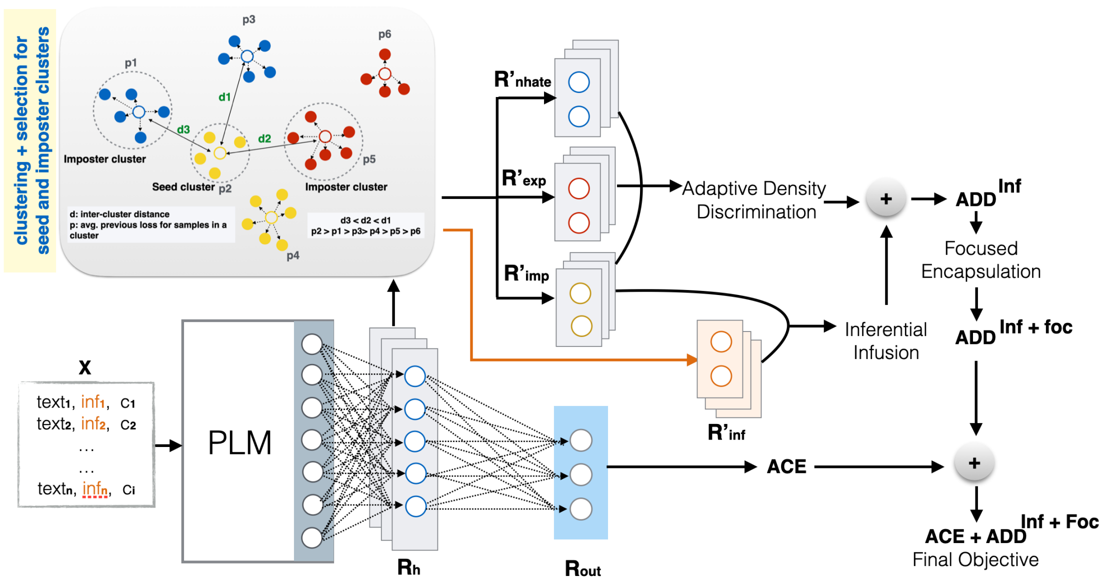


Figure 4.4: The architecture of `FiADD`. Input X is a set of texts, implied annotations (only for implicit class), and class labels. PLM (frozen). R'_{nhate}, R'_{exp}, and R'_{imp} are the representatives for seed and imposter clusters of non-hate, explicit, and implicit, respectively. R'_{inf} represents inferential meaning for corresponding R'_{imp}. ACE is alpha cross-entropy, and $ADD^{Inf+foc}$ is the ADD with inferential + focal objective.

4.3.6 Design of `FiADD`

The proposed **F**ocused **I**nferential **A**daptive **D**ensity **D**iscrimination (`FiADD`) framework consists of a standard finetuning pipeline with encoder-only PLM followed by a projection layer R_h and a classification head (CH). To reduce the distance between the implicit hate (imp) and implied clusters (inf), `FiADD` measures the average distance of implicit points from implied meaning as a ratio of its distance to explicit and non-hate subspaces. During the PLM finetuning, our setup combines with the CE loss to improve the detection of hate. An overview of `FiADD`'s architecture is reflected in Figure 4.4. For each training instance $(x_d, y_d) \in X$, with x_d input post and y_d label, $x_p = PLM(x_d)$ is the encoded representation obtained from the PLM. The encodings are projected to obtain $r_d = R_h(x_p)$. Here $x_p \in \mathbb{R}^{768}$ and $r_d \in \mathbb{R}^{128}$ as $r_d \ll x_d$ allows for faster clustering.

Inferential infusion. As each output label y_d belongs to one of the distinct classes ($c_i \in C$), we employ the respective embeddings r_d and offline K-means algorithm to obtain K subclusters per class. For implicit hate samples, the latent representation of their implied/inferential counterparts $\tilde{x_d}$ is denoted as $\tilde{r_d} = R_h(PLM(\tilde{x_d}))$. If $r_1^m, \ldots, r_d^m$ are representations for D samples of mth implicit cluster, then $\tilde{r}_1^m, \ldots, \tilde{r}_d^m$ represent their respective inferential forms. The updated **I**nferential **A**daptive **D**ensity **D**iscrimination (ADD^{inf}) help reduce the distance between $(r_d, \tilde{r_d})$ for implicit hate samples via Equation 4.3.

$$p^{ADD^{inf}}(r_d^m) = \frac{e^{-\frac{1}{2\sigma^2}\|r_d^m - \mu^m\|_2^2 - \alpha} + e^{-\frac{1}{2\tilde{\sigma^2}}\|\tilde{r_d^m} - \tilde{\mu^m}\|_2^2 - \alpha}}{\sum_{\mu^o : C(\mu^o) \neq C(r_m^d)} e^{-\frac{1}{2\sigma^2}\left\|r_d^m - \mu^o\right\|_2^2}} \tag{4.3}$$

Here, μ^m (σ^2) and $\tilde{\mu^m}$ ($\tilde{\sigma^2}$) are the mean (variance) representations of the implicit and inferential/implied form for the mth implicit cluster, respectively. Equation 4.1 can be broken into two additive parts. The first part is equivalent to ADD thus focusing on reducing the intra-cluster distance within the implicit class. The second part brings the implicit class closer to its implied meaning. Meanwhile, in the case of explicit or non-hate clusters, there is no mapping to the inferential/implied cluster, and ADD^{inf} in Equation 4.3 reduces to ADD in Equation 4.1.

Focal weight. Both ADD^{inf} and ADD assign uniform weight to all samples under consideration. In contrast, we establish (Section 4.3.4) that some instances are closer to the boundary of the imposter clusters and more challenging to classify, i.e., contribute more to the loss. Inspired by the concept of focal loss (Lin *et al.*, 2017), we improve the ADD^{inf} objective by introducing $ADD^{inf+foc}$. Under $ADD^{inf+foc}$, the loss on each sample is multiplied by a factor called the *focused term* $(1 - p^{ADD^{inf}}(r_d^m))^{\gamma}$. γ, a hyperparameter, acts as a magnifier. The formulation assigns uniform weight as $\gamma \to 0$, reducing to ADD^{inf}. Analogously, the focal term is paying "more attention" to specific data points. Even without inferential infusion, our novel focal term can be incorporated as ADD^{foc} as enlisted in Equation 4.4.

$$\ell^{ADD^*}(\Theta) = \frac{1}{MD} \sum_{m=1}^{M} \sum_{d=1}^{D} \left[-(1 - p^{ADD^*}(r_d^m))^{\gamma} log(p^{ADD^*}(r_d^m)) \right] \tag{4.4}$$

Here, $\ell^{ADD^{inf+foc}}$ ($\ell^{ADD^{foc}}$) captures the setup with (without) inferential objective.

We utilise $p^{ADD^{inf}}$ (Equation 4.3) for the former and p^{ADD} (Equation 4.1) for the latter. Despite ADD^{foc} being a minor update on ADD, we empirically observe that focal infusion improves ADD. Our novel inclusion of focal ADDcan thus be extended to other classification tasks, especially in computer vision.

Overall loss. It should be noted that selecting the seed cluster and its subsequent imposter clusters is a random process for initial iterations. We assign the label with the highest loss margin for later iterations as the seed. Apart from employing the embedding r_d in ADD^{inf}, it also passes through a CH $CH(r_d)$. We combine CE with the focal inference via Equation 4.5, with β controlling the contribution of the two components.

$$\ell(\Theta) = \beta\ell^{CE}(\Theta) + (1-\beta)\ell^{ADD^*}(\Theta) \tag{4.5}$$

Inference. During inference, the system does not have access to implied meaning. Once the PLM is trained via `FiADD`, the CH performs similarly to any finetuned model. Here, we rely on the latent space being modified so that the implicit statements are closer to their semantic or implied form.

Note on K-means. As a clustering algorithm, K-means is the most generic as it does not assume any dataset property (like hierarchy) except for the semantic similarity of the samples. Further, the K-means computation happens offline in each epoch, i.e., it does not consume GPU resources. In the future, we aim to employ faster versions of K-means to improve training latency. Meanwhile, the computational complexity of `FiADD` during inference is the same as the finetuned PLM.

4.3.7 Experimental setup

In this section, we enlist the performance for classifying implicit hate and the model's generalisability measured via macro-F1.

PLMs. For the task of hate speech detection, we begin with BERT (Devlin *et al.*, 2019), which is the most widely used PLM for downstream NLP tasks, including hate speech. We further employ a domain-specific HateBERT (Caselli *et al.*, 2021*a*) model to establish generalisability beyond BERT embedding. The PLM variants are 'bert-base-uncased' for BERT and 'GroNLP/hateBERT' for HateBERT.

Hyperparameters. We run all experiments on two Nvidia V100 GPUs. Three random seeds (1, 4, 7) are used per setup. We report each setup's best performance based on overall macro-F1 out of three random seeds, where the best seed for a setup may vary. We follow an 80-20 split for the dataset across experiments (specific to the seed). In initial experiments, we observe that $ADD^{inf+foc}$ has a stronger influence on later iterations, whereas CE influences the initial ones. Thus, to balance them throughout the training process, we put equal weightage on both using $\beta = 0.5$. We consider $K = 3$ with $M = 2$ impostors for all experiments. We leave the experiments for β and M for future work. We set 100 as the maximum K-means iterations in each training step. During finetuning, each training cycle is executed for a maximum of 5000 epochs with all layers of PLM frozen.

Baselines. As ADD (Rippel *et al.*, 2016) already established its efficacy over other distance metrics, we focus on those setups that compare various ADD formulations

Table 4.6: Baseline selection based on: (a) Improvements of ADD^{foc} over vanilla ADD for two-way hate speech classification via LSTM. (b) Standalone Comparison of ADD^{foc} with ACE for three-way hate speech classification via BERT. We report each model's best performance based on overall macro-F1 out of three random seeds.

(a)

Dataset	Metric (F1)	ADD	α-ADD	ADD^{foc}
LatentHatred	Macro	0.557	0.601	**0.665**
	N-Hate	0.646	0.695	0.782
	Hate	0.469	0.509	0.629
ImpGab	Macro	0.554	0.566	**0.639**
	N-Hate	0.919	0.911	0.927
	Hate	0.189	0.222	0.351
AbuseEval	Macro	0.552	0.554	**0.634**
	N-Hate	0.820	0.780	0.851
	Hate	0.284	0.327	0.417

(b)

Dataset	Metric (F1)	ADD^{foc}	ACE
LatentHatred	Macro	0.457	**0.533**
	N-Hate	0.690	0.772
	EXP	0.221	0.268
	IMP	0.462	0.558
ImpGab	Macro	0.451	**0.470**
	N-Hate	0.926	0.924
	EXP	0.347	0.390
	IMP	0.080	0.097
AbuseEval	Macro	0.504	**0.530**
	N-Hate	0.817	0.854
	EXP	0.491	0.503
	IMP	0.203	0.233

instead. Further, we evaluate varying combinations of CE and our proposed losses.

- **ADD vs. ADD$^{\mathbf{foc}}$.** We assess the improvement in the performance of ADD^{foc} over vanilla ADD (Equation 4.4) without the influence of CE. We follow the same prediction setup adopted by Rippel *et al.* (2016), where a sample gets assigned the label based on the nearest cluster in the trained latent space during inference. We choose a simple LSTM (Hochreiter and Schmidhuber, 1997) model for quicker experimentation and compare the original ADD formulation with class weighted ADD (α-ADD) and our proposed ADD^{foc}.
- **CE vs ADD$^{\mathbf{foc}}$.** Further, we perform a 3-way classification using BERT to compare standalone ACE (α CE) against standalone ADD^{foc}.

Table 4.6(a) shows a significant performance improvement of 8.2-10.8% in overall macro-F1 using ADD^{foc} across all three datasets. We thus recommend using our ADD^{foc} variant instead of vanilla ADD for future works. Interestingly, we note that α-ADD does not outperform ADD^{foc}. Hence, it is not employed in further experiments. From the results in Table 4.6(b), we observe that ACE outperforms the standalone ADD^{foc} by a substantial margin of 7.6%, 1.9%, and 2.6% for `LatentHatred`, `ImpGab`, and `AbuseEval` respectively. Based on the above two experiments, we employ ACE and ACE+ADD^{foc} as our comparative systems.

Note on generalisability testing. Apart from hate speech, we also assess the utility of `FiADD` on other NLP tasks where implied and surface forms can differ. We specifically evaluate the use cases for irony, sarcasm, and stance detection. The experimental setup and results are detailed in Appendix E. Succinctly speaking, we see the performance of these tasks improve under `FiADD`.

4.3.8 Results

In this section, we discuss the performance comparison of `FiADD`, alongside the impact of variations in PLMs, hyperparameters, and random seeding. Under both two-way

Table 4.7: Results for two-way hate classification on BERT and HateBERT. N-Hate: non-hate and Hate: implicit + explicit hate. For each model under comparison, we report its best performance based on overall macro-F1 out of three random seeds $(1, 4, 7)$. We also highlight the highest Hate class macro-F1 that the respective model can achieve.

Dataset	Metric (F1)	BERT			HateBERT		
		ACE	ACE+ADD^{foc}	ACE+ADD$^{inf+foc}$	ACE	ACE+ADD^{foc}	ACE+ADD$^{inf+foc}$
`LatentHatred`	Macro	0.6991	**0.7049**	0.7039	0.7121	**0.7135**	0.7121
	N-Hate	0.7599	0.7603	0.7638	0.7843	0.7787	0.7791
	Hate	0.6383	**0.6495**	0.6439	0.6400	**0.6484**	0.6450
	Highest Hate F1	0.6478	0.6491	**0.6548**	0.6615	0.6572	**0.6630**
`ImpGab`	Macro	0.6709	0.6886	**0.6956**	0.6889	**0.7027**	0.6987
	N-Hate	0.9171	0.9313	0.9355	0.9239	0.9398	0.9344
	Hate	0.4247	0.4459	**0.4556**	0.4538	**0.4657**	0.4631
	Highest Hate F1	0.4281	0.4494	**0.4562**	0.4538	0.4663	**0.4631**
`AbuseEval`	Macro	0.7075	**0.7131**	0.7124	0.7189	0.7176	**0.7202**
	N-Hate	0.8729	0.8779	0.8675	0.8767	0.8711	0.8808
	Hate	0.5420	0.5483	**0.5574**	0.5610	**0.5641**	0.5595
	Highest Hate F1	0.5502	0.5583	**0.5625**	0.5642	**0.5710**	0.5691

Table 4.8: Results for two-way hate classification on BERT and HateBERT. N-Hate: non-hate, EXP: explicit hate, and IMP: implicit hate. For each model under comparison, we report its best performance based on overall macro-F1 out of three random seeds.

Dataset	Metric (F1)	BERT			HateBERT		
		ACE	ACE+ADD^{foc}	ACE+ADD$^{inf+foc}$	ACE	ACE+ADD^{foc}	ACE+ADD$^{inf+foc}$
`LatentHatred`	Macro	0.5327	0.5331	**0.5336**	0.5537	**0.5607**	0.5571
	N-Hate	0.7722	0.7609	0.7474	0.7480	0.7654	0.7829
	EXP	0.2682	0.2775	0.2776	0.3403	0.3300	0.3111
	IMP	0.5577	0.5610	**0.5759**	0.5728	**0.5866**	0.5774
`ImpGab`	Macro	0.4621	0.4575	**0.4668**	0.4797	0.4772	**0.4813**
	N-Hate	0.9198	0.9197	0.9258	0.9230	0.9434	0.9339
	EXP	0.3775	0.3850	0.3819	0.4055	0.4076	0.4190
	IMP	0.0889	0.0678	**0.0928**	**0.1105**	0.0806	0.0909
`AbuseEval`	Macro	0.5300	0.5328	**0.5398**	0.5309	0.5311	**0.5313**
	N-Hate	0.8541	0.8370	0.8867	0.8532	0.8611	0.8606
	EXP	0.5027	0.5256	0.5234	0.5038	0.5138	0.5111
	IMP	0.2333	**0.2359**	0.2094	**0.2357**	0.2185	0.2222

and three-way classification, `FiADD` is applied to the last hidden layer, assuming three classes in the dataset.

Two-way hate classification. From Table 4.7, we note that `FiADD` variants improves overall macro-F1 by 0.58 ($\uparrow$ 0.83%), 2.47 ($\uparrow$ 3.68%), and 0.56 ($\uparrow$ 0.79%) in `LatentHatred`, `ImpGab`, and `AbuseEval`, respectively using BERT. However, except for maximising hate macro-F1, the inferential objective does not significantly impact the final macro-F1 in the case of a two-way classification. It can be explained by the partially conflicting objectives between the final two-way result and $ADD^{inf+foc}$'s three-way objective, leading to higher misclassification.

Three-way hate classification. Inferential infusion reasonably impacts the outcome of the three-way classification task (Table 4.8). Overall, in 3-way classification, $ADD^{inf+foc}$ provide an improvement of 0.09 ($\uparrow$ 0.17%), 0.47 ($\uparrow$ 1.02%), and 0.98 ($\uparrow$ 1.85%) in macro-F1 for `LatentHatred`, `ImpGab`, and `AbuseEval`, respectively, on BERT. *It is noteworthy that we observe an even higher level of improvement for the implicit hate class than overall.* Compared to ACE in three-way classification, ADD^{foc} helps `AbuseEval` with an improvement of 0.26 macro-F1 ($\uparrow$ 1.11%) in implicit hate. Meanwhile, $ADD^{inf+foc}$ helps `LatentHatred` and `ImpGab` with an improvement of 1.82 ($\uparrow$ 3.26%) and 0.39 ($\uparrow$ 4.39%), macro-F1 respectively, in implicit hate.

Impact of domain-specific PLM. Under HateBERT, `FiADD` variants improve two-way classification by overall 0.14 ($\uparrow$ 0.20%), 1.38 ($\uparrow$ 2.00%) and 0.13 ($\uparrow$ 0.18%) for `LatentHatred`, `ImpGab`, and `AbuseEval` respectively. Similarly, `FiADD` variants improve three-way classification by overall 0.7 ($\uparrow$ 1.26%), 0.16 ($\uparrow$ 0.34%), 0.04 ($\uparrow$ 0.08%) for `LatentHatred`, `ImpGab`, and `AbuseEval` respectively. However, the results with HateBERT show more variability. While all datasets in two-classification via HateBERT benefit from `FiADD` implicitness of `AbuseEval`, and `ImpGab` suffer under three-way classification. This variation can be attributed to a lot more offensive and slur terms in HateBERT's training than in BERT. Through this analysis, we are able to comment on the domain-specific (HateBERT) vs general-purpose (BERT) systems and their role in finetuning.

Seed-wise analysis. Across three random seeds, two PLMs, and three datasets, we record the performance for 18 setups, each in two-way and three-way hate speech detection. We note from Table 4.9 that across two and three-way, out of the 36 combinations, only four instances register a drop in performance. It corroborates that `FiADD`'s improvements are not limited to a specific initialisation setup. Interestingly, the setups that register failure are all under HateBERT.

> The above observations further call into question the role of domain-specific PLMs, as pointed out in the last chapter's work (Masud *et al.*, 2024*b*).

Significance of hyperparameters. We further experiment with the hyperparameters of the `FiADD`. The experiments are performed on a two-way hate classification task on the `AbuseEval` dataset using BERT. The limited range for the probe is heuristically defined based on the sample size of categories. We recommend determining the values on a case-by-case basis for optimised performance. Figure 4.5(a) represents the significance of the number of subclusters per class (k) in the range of [2-4]. We observe comparable performance for $K = 3$ or 4 and settle for $K = 3$. The intuition is that within a subcluster of a class, the three subclusters represent a case of one of them having a high affinity to the class itself and two others being closer to their imposter classes. For example, within the implicit hate class, we assume at least one subcluster is easy to label as implicit, while there will likely be at least one cluster each that is closer to explicit and non-hate classes. Consequently, the setup leads to an imposter cluster value of $M = 2$.

Meanwhile, the significance of the γ coefficient used in the focused objective is presented in Figure 4.5(b). The probe is limited to [1-5] with a unit interval as followed in existing literature (Lin *et al.*, 2017). We observe the best outcome with $\gamma = 2$, which incidentally aligns with the best value identified by Lin *et al.* (2017).

4.3.9 Error analysis

The motivation for `FiADD` is that implicit is closer to non-hate than explicit hate. Employing `FiADD` should correct the misclassified implicit labels if this hypothesis holds. On the other hand, a false-positive may occur if the example is already close to the explicit subspaces. Further, moving it toward explicit space can cause misclassification. We, thus, consider a positive/negative case where the predicted label for an implicit sample is correctly/incorrectly classified. To explain these two scenarios, we estimate

Table 4.9: Seed-wise results for (a) two-way and (b) three-way classification across all three hate-speech datasets using two PLMs, BERT and HateBERT. Highlighted with green colour are the outcomes where one of the variants of `FiADD` outperforms baseline ACE.

(a) 2-way

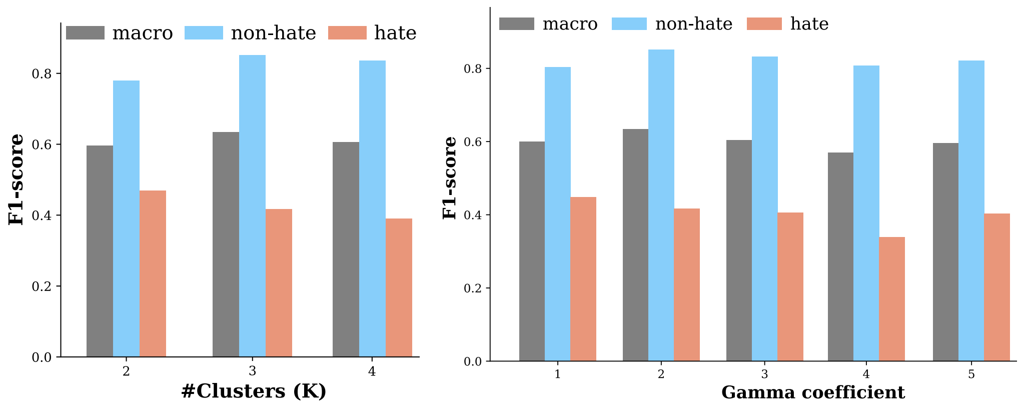
Dataset	Seed	Metric (F1)	BERT ACE	BERT $\mathrm{ACE + ADD^{foc}}$	BERT $\mathrm{ACE + ADD^{inf+foc}}$	HateBERT ACE	HateBERT $\mathrm{ACE + ADD^{foc}}$	HateBERT $\mathrm{ACE + ADD^{inf+foc}}$
`LatentHatred`	1	Macro	0.69658	0.70291	0.69780	0.70843	0.69983	0.70143
		N-Hate	0.76353	0.76010	0.76069	0.77514	0.75034	0.76016
		Hate	0.62962	0.64571	0.63492	0.64171	0.64931	0.64271
	4	Macro	0.69388	0.68951	0.69611	0.70720	0.70423	0.70709
		N-Hate	0.75595	0.75246	0.77109	0.76370	0.75511	0.76389
		Hate	0.63182	0.62656	0.62113	0.65070	0.65335	0.65030
	7	Macro	0.69907	0.70490	0.70387	0.71212	0.71353	0.71205
		N-Hate	0.75987	0.76033	0.76384	0.78429	0.77868	0.77907
		Hate	0.63827	0.64946	0.64389	0.63995	0.64838	0.64502
`ImpGab`	1	Macro	0.67087	0.68862	0.69557	0.68886	0.69790	0.69872
		N-Hate	0.91708	0.93134	0.93551	0.92391	0.92951	0.93438
		Hate	0.42467	0.44590	0.45563	0.45380	0.46630	0.46307
	4	Macro	0.66411	0.67941	0.67591	0.68248	0.67406	0.68307
		N-Hate	0.91361	0.92403	0.91952	0.92069	0.91105	0.92318
		Hate	0.41461	0.43478	0.43231	0.44428	0.43708	0.44295
	7	Macro	0.66402	0.68237	0.68206	0.68720	0.70271	0.68962
		N-Hate	0.91019	0.92933	0.92782	0.92266	0.93976	0.93049
		Hate	0.41784	0.43541	0.43630	0.45175	0.46565	0.44874
`AbuseEval`	1	Macro	0.69253	0.70627	0.70502	0.70633	0.71030	0.70699
		N-Hate	0.85193	0.86709	0.87395	0.87774	0.87652	0.88362
		Hate	0.53313	0.54545	0.53609	0.53492	0.54409	0.53035
	4	Macro	0.70748	0.71313	0.71244	0.71885	0.71761	0.72018
		N-Hate	0.87293	0.87792	0.86754	0.87667	0.87112	0.88082
		Hate	0.54204	0.54833	0.55735	0.56103	0.56410	0.55953
	7	Macro	0.67794	0.68587	0.68944	0.69350	0.70146	0.69910
		N-Hate	0.85393	0.86020	0.85694	0.85240	0.85052	0.86129
		Hate	0.67794	0.68587	0.68944	0.69350	0.70146	0.69910

(b) 3-way

Dataset	Seed	Metric (F1)	BERT ACE	BERT $\mathrm{ACE + ADD^{foc}}$	BERT $\mathrm{ACE + ADD^{inf+foc}}$	HateBERT ACE	HateBERT $\mathrm{ACE + ADD^{foc}}$	HateBERT $\mathrm{ACE + ADD^{inf+foc}}$
`LatentHatred`	1	Macro	0.50683	0.51872	0.51727	0.53277	0.53023	0.52951
		N-Hate	0.76214	0.76201	0.74872	0.76216	0.75547	0.76122
		EXP	0.20159	0.25344	0.25296	0.25575	0.26912	0.26109
		IMP	0.55678	0.54072	0.55013	0.58041	0.56609	0.56622
	4	Macro	0.52340	0.52746	0.52819	0.55372	0.56065	0.54937
		N-Hate	0.77192	0.75288	0.76089	0.74803	0.76535	0.75695
		EXP	0.26631	0.25862	0.28870	0.34032	0.33000	0.32524
		IMP	0.53196	0.57088	0.53499	0.57281	0.58660	0.56591
	7	Macro	0.53267	0.53312	0.53361	0.54265	0.54197	0.55713
		N-Hate	0.77215	0.76094	0.74736	0.76769	0.77002	0.78290
		EXP	0.26815	0.27748	0.27762	0.29891	0.28428	0.31111
		IMP	0.55772	0.56095	0.57587	0.56136	0.57162	0.57738
`ImpGab`	1	Macro	0.46284	0.45740	0.47635	0.47448	0.47417	0.47325
		N-Hate	0.90876	0.91715	0.92309	0.92432	0.92950	0.91882
		EXP	0.36140	0.38783	0.37983	0.39767	0.40897	0.40336
		IMP	0.11834	0.06722	0.12612	0.10144	0.08403	0.09756
	4	Macro	0.45289	0.45772	0.43606	0.46788	0.47415	0.44959
		N-Hate	0.92661	0.91054	0.91243	0.92351	0.92469	0.91982
		EXP	0.38078	0.36646	0.37195	0.38563	0.42696	0.40542
		IMP	0.05128	0.09615	0.02380	0.09448	0.07079	0.02352
	7	Macro	0.47048	0.45741	0.46351	0.47969	0.47721	0.48128
		N-Hate	0.92416	0.93149	0.93998	0.92302	0.94337	0.93388
		EXP	0.39033	0.40073	0.38159	0.40554	0.40761	0.41904
		IMP	0.09696	0.04000	0.06896	0.11049	0.08064	0.09090
`AbuseEval`	1	Macro	0.51761	0.52082	0.52249	0.52462	0.52638	0.52559
		N-Hate	0.84885	0.84862	0.849518	0.86772	0.87497	0.87748
		EXP	0.49009	0.51483	0.51033	0.53555	0.52680	0.52727
		IMP	0.21390	0.19900	0.20765	0.17058	0.17737	0.17204
	4	Macro	0.53002	0.53283	0.53982	0.52609	0.52095	0.52764
		N-Hate	0.85409	0.83699	0.88670	0.86536	0.86110	0.87867
		EXP	0.50272	0.52564	0.52338	0.52133	0.50485	0.51759
		IMP	0.23325	0.23587	0.20938	0.19158	0.19689	0.18666
	7	Macro	0.51914	0.52951	0.52534	0.53091	0.53110	0.53129
		N-Hate	0.84480	0.85799	0.85942	0.85322	0.86106	0.86060
		EXP	0.50420	0.51487	0.51041	0.50383	0.51378	0.51106
		IMP	0.20843	0.21568	0.20618	0.23569	0.21848	0.22222

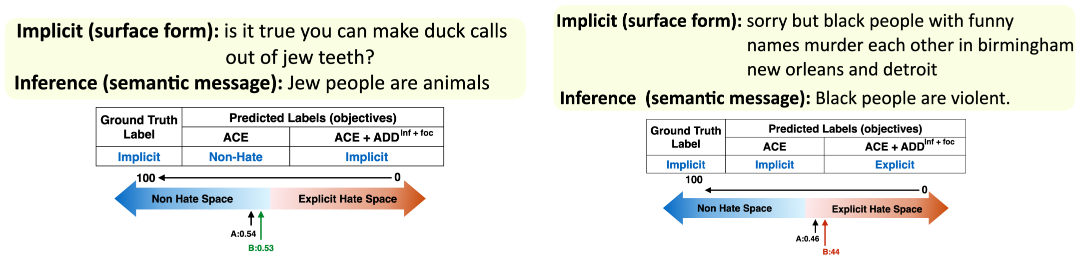
the relative distance of the implicit sample from explicit and non-hate clusters. First, we perform K-means clustering on non-hate and explicit latent spaces to identify their centres. We then calculate the average Manhattan distance between the implicit samples and these local density centres. Finally, we obtain the relative score from explicit space by normalising the average explicit distance by the sum of average distances from non-hate and explicit spaces. We highlight a positive and a negative case in Figure 4.6.

In the positive case, Figure 4.6(a), the implicit sample is closer to the non-hate space (Point A) under the ACE objective. After employing the `FiADD`, its relative position moves away from non-hate and closer to explicit (point B). In contrast, for the negative case, Figure 4.6(b), where the implicit sample is initially close to explicit hate (point A), our objective leads to misclassification. In the future, this problem can be reduced by



(a) (b)

Figure 4.5: The variation in performance with changing values of (a) number of clusters (k) and (b) focal parameter (γ). We employ BERT on `AbuseEval` with ADD^{foc} in the two-way classification.



(a) Positive case (b) Negative case

Figure 4.6: Error analysis with (a) correctly and (b) incorrectly classified samples in three-way classification on `LatentHatred`. Here, scores A and B are the relative positions of implicit sample w.r.t non-hate and explicit space, finetuned with ACE and $ADD^{inf+foc}$, respectively.

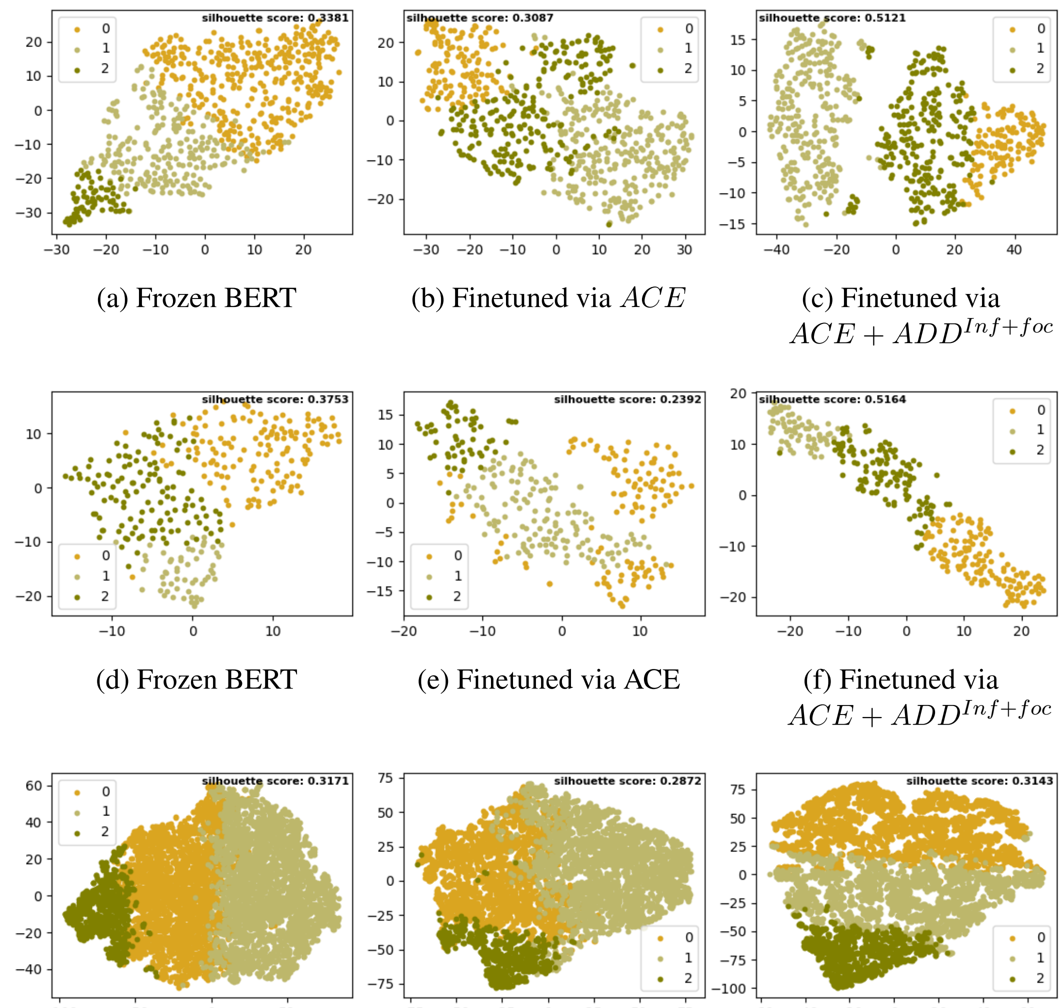
introducing a fixed constraint on the distance between implicit and explicit intact.

4.3.10 Does **`FiADD`** really improve implicit hate detection?

Given that the overall macro-F1 results on hate speech detection vary in a narrow range, significance testing will be inconclusive. Building upon the cluster assessment in the error analysis (Section 4.3.9), where we examine only a single positive and negative sample, we now perform an overall evaluation of how $ADD^{inf+foc}$ manipulates the embedding space. Inspired by the existing literature examining the latent space for hate speech embeddings (Fortuna *et al.*, 2020; Kim *et al.*, 2022; Ocampo *et al.*, 2023c), we attempt to quantify the inter-cluster separation via Silhouette scores (SS).

Silhouette score. It measures the 'goodness' of the clustering technique. It is calculated as a tradeoff between within-cluster similarity and inter-cluster dissimilarity. Consider a system with E points (e_i), each point belonging to one of the N clusters $c^j(e_i)$. For $e_i \in c^a$, its Silhouette score $SS_i = \frac{max(p_i,q_i)}{q_i - p_i}$. p_i captures the intra-cluster distance of e_i to all the points within the cluster it belongs; $p_i = \frac{1}{|c^a|-1}\sum_{e^j \in c^a} dist(e_i, e_j)$. q_i captures the inter-cluster distance of $e_i \in c^a$ to all the points in the nearest cluster to c^a; $q_i = \frac{1}{|c^b|}\sum_{e^j \in c^b} dist(e_i, e_j)$. The Silhouette score of a setup is, thus, $SS =$

$\frac{1}{|E|} \sum_{e_i \in E} SS_i$. Silhouette scores are measured on a scale of -1 to 1, with -1 being the worst set of cluster assignments.



(a) Frozen BERT (b) Finetuned via ACE (c) Finetuned via $ACE + ADD^{Inf+foc}$

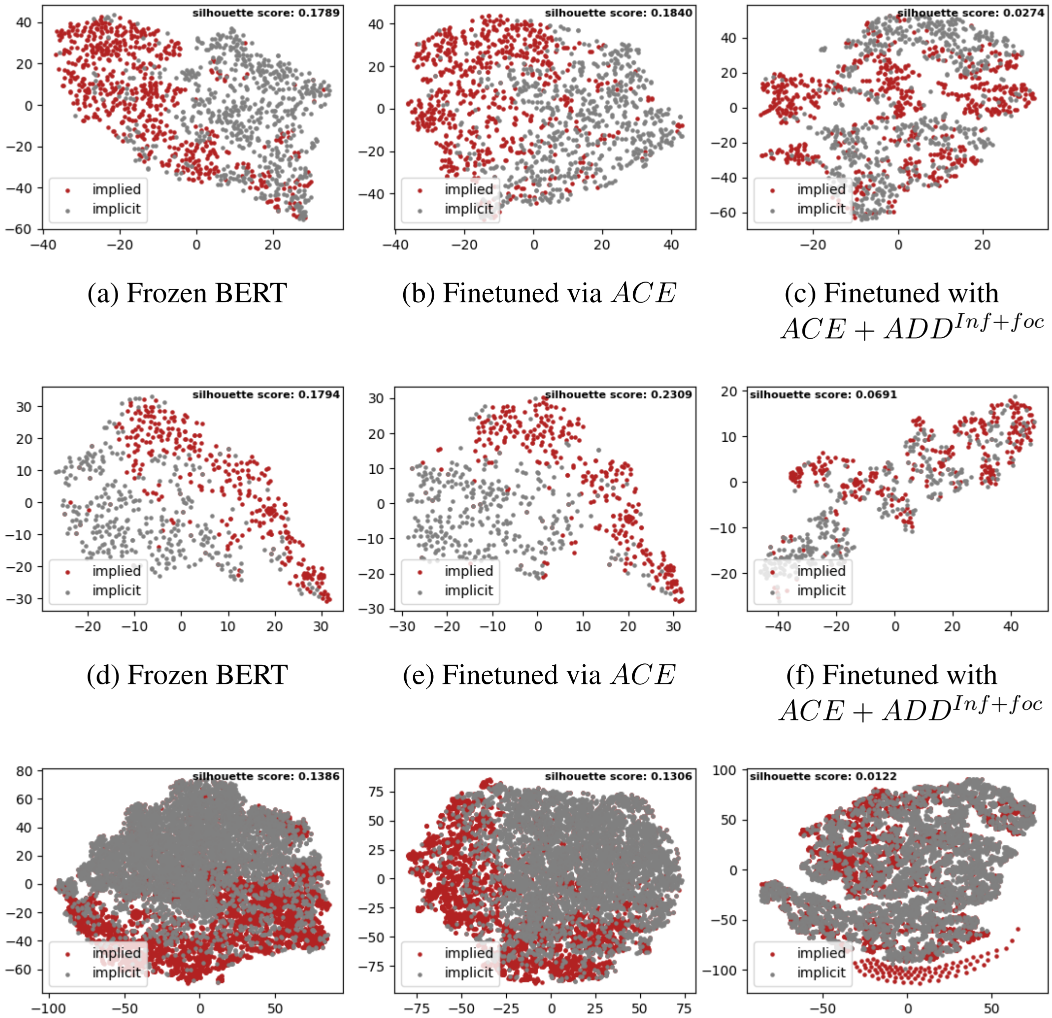
(d) Frozen BERT (e) Finetuned via ACE (f) Finetuned via $ACE + ADD^{Inf+foc}$

(g) Frozen BERT (h) Finetuned via ACE (i) Finetuned via $ACE + ADD^{Inf+foc}$

Figure 4.7: 2D t-SNE plots of the last hidden representations after applying K-means (K=3) on the implicit class for `AbuseEval` (a, b, c), `ImpGab` (d, e, f) and `LatentHatred` (g, h, i). $\{0, 1, 2\}$ are the subcluster ids. The higher the Silhouette score, the better the discrimination between the clusters.

Subclustering objective. After applying the $ADD^{inf+foc}$ objective, we expect not only the per-class clusters to be sufficiently separated but also the subclusters in each class to be better segregated to match their local neighbourhood better. Figure 4.7 shows the implicit embedding space of `AbuseEval`, `ImpGab` and `LatentHatred` after applying K-means on the frozen BERT embedding (a, d, g), BERT finetuned with the ACE (b, e, h), and `FiADD` (c, f, i) on three-way hate classification. The higher the Silhouette score, the better the subclusters are separated. 0.34, 0.31, and 0.51 are the scores for cases (a), (b), and (c), respectively, in `AbuseEval`. 0.38, 0.24, and 0.52 are the scores for cases (d), (e), and (f), respectively, in `ImpGab`. 0.32, 0.29, and 0.32 are the scores for cases (d), (e), and (f), respectively, in `LatentHatred`. Consequently, an increase of 0.20, 0.28, and 0.03 scores is observed when comparing `FiADD` with BERT+ACE for `AbuseEval`, `ImpGab`, and `LatentHatred` respectively. Interestingly, for `LatentHatred` the score does not improve over the frozen BERT, even though it improves over BERT+ACE. A deeper analysis with multiple K values might help.

This increase in scores validates that the local densities within a class get further refined under $ADD^{inf+foc}$ objective. As expected, ACE sub-optimally treats the implicit class as a single homogeneous cluster.



(a) Frozen BERT (b) Finetuned via ACE (c) Finetuned with $ACE + ADD^{Inf+foc}$

(d) Frozen BERT (e) Finetuned via ACE (f) Finetuned with $ACE + ADD^{Inf+foc}$

(g) Frozen BERT (h) Finetuned via ACE (i) Finetuned with $ACE + ADD^{Inf+foc}$

Figure 4.8: 2D t-SNE plots of the last hidden representations obtained for the implicit class and its respective inferential (implied) set for `AbuseEval` (a, b, c), `ImpGab` (d, e, f) and `LatentHatred` (g, h, i). The lower the Silhouette score, the closer the surface and implied forms of hate.

Inferential infusion. Given that $ADD^{inf+foc}$ brings the surface and semantic forms of implicit hate closer, we expect a significant drop in Silhouette scores between these clusters under `FiADD`. Figure 4.8 visualises the embedding space of frozen BERT (a, d, g), BERT finetuned with the ACE (b, e, h), and `FiADD` (c, f, i) on three-way classification for `AbuseEval`, `ImpGab`, and `LatentHatred`. 0.18, 0.18, and 0.03 are the scores for cases (a), (b), and (c), respectively, in `AbuseEval`. 0.18, 0.23, and 0.07 are the scores for cases (d), (e), and (f), respectively, in `ImpGab`. 0.14, 0.13, and 0.01 are the scores for cases (g), (h), and (i), respectively, in `LatentHatred`. It is essential to highlight that for both frozen BERT and BERT+ACE, there is no explicit objective to bring the implicit and implied clusters together. Hence, they act as a baseline for comparing how well the $ADD^{inf+foc}$ objective brings the two spaces closer. A drop of 0.15, 0.16, and 0.12 in the Silhouette score is observed when comparing BERT+ACE with `FiADD` for `AbuseEval`, `ImpGab`, and `LatentHatred`, respectively.

The analysis corroborates that the implicit and implied meaning representations are brought significantly closer to each other by employing our model. The latent space analysis also quantifies our manual annotations for `AbuseEval` *and* `ImpGab`*, as inferential infusion is improving the detection of implicitness.*

4.3.11 Limitations and future works

- **Dependency on manual annotations.** First and foremost, the current setup utilises manual annotations of implicit meaning to be available for inferential clustering. In the future, we expect an infusion of generative models to pseudo-annotate the implied meaning, which can be paraphrased and rectified by human annotators on a need basis. Further, the proposed setup can be employed as an external loss to nudge the generation of better-quality adversarial examples.
- **More efficient clustering.** The proposed setup, being a novel approach in the direction of implicit detection, works on the de facto K-means and uses the same number of subclusters for all datasets. Recent advancements in hashing and dictionary techniques can improve computational efficiency.

4.4 A survey of bias mitigation in toxicity detection

4.4.1 Motivation

Research gaps. Bias mitigation methods in NLP have motivated researchers to apply bias mitigation for toxicity detection (Garrido-Muñoz *et al.*, 2021; Blodgett *et al.*, 2020; Weidinger *et al.*, 2021). However, the results are not as straightforward as expected. The argument goes back to the subjective vs. objective nature of tasks like toxicity detection vs. textual entailment. At the same time, several surveys have examined the methods for toxicity detection and highlighted a lack of bias mitigation without surveying the latter (Schmidt and Wiegand, 2017; Fortuna and Nunes, 2018). Meanwhile, Yin and Zubiaga (2021) surveyed the literature addressing the robustness of hate speech detection methods and discussed the subject of bias in hate speech detection. While their discussion remained general commentary, there is still a lack of an extensive understanding of these methods. It inspires us to conduct a survey that can help better understand the current state of bias mitigation in toxicity detection literature.

Research questions. Within the plethora of work analysing bias mitigation in toxicity detection, we aim to uncover the following:

RQ1: What taxonomy of bias best suits the toxicity literature?
RQ2: What are the key categories of bias in toxicity literature, and how does it correlate to the NLP pipeline?

Contribution summary. We develop a taxonomy of bias based on the sources and targets of harm (Garg *et al.*, 2023). Each bias mitigation method can be applied to one or more data transformation stages of an ML pipeline (Suresh and Guttag, 2021). As a part of our literature survey, we review the reproducibility of existing debiasing methods employed for toxicity detection. As a byproduct of reproducing existing results, we discover a compelling phenomenon of bias shift in knowledge generalisation-based methods and provide a short description of the same.

Note on survey methodology. Following Yin and Zubiaga (2021), we consider Google Scholar as the primary search engine for curating relevant papers. We focus on toxicity debiasing studies done within the last five years and mainly look at research published in 2016 onwards. We start with relevant keywords such as "bias", "toxic speech", "abusive speech", and "hate speech", shortlisting a seed set of papers through their abstracts. Papers are also collected from recent proceedings of relevant data mining, NLP, and web-science conferences (ACL, EMNLP, NAACL, AAAI, WebSci, ICWSM, etc.), journals (TACL, TKDD, PLOS, etc.), and workshops (WOAH, etc.). We also visit the cited papers of the seed papers to locate relevant papers further. This shortlisting process is done between September and November 2021.

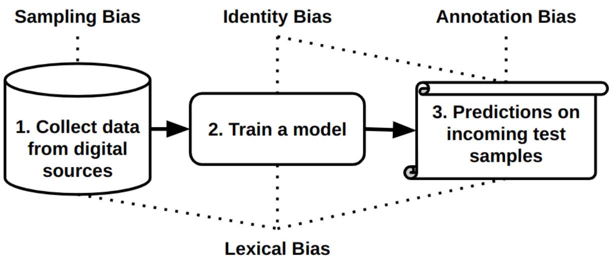


Figure 4.9: The pipeline of training and evaluation of NLP models can be visualised as a sequence of (i) data collection, (ii) training of models, and (iii) employing the models on incoming test samples. Here, collecting data from digital sources can introduce sampling and lexical biases (if specific web pages/digital footprints are ignored). Model training can also introduce linguistic bias when frequently picking co-occurring phrases. Training processes and skewed datasets can introduce identity-based prejudice (race, gender, age, political affiliation, etc.). If predictions on downstream NLP tasks happen on human-labelled gold datasets, it can be a source of annotation and sampling biases (not depicted here). Both linguistic and identity-based biases manifest during the model evaluation and in-the-wild testing.

4.4.2 Categories of bias

In addition to the irreducible bias, all machine learning models, including toxicity detection, assume some data bias to perform predictions (Bolukbasi *et al.*, 2016; Caliskan *et al.*, 2017; Dixon *et al.*, 2018). However, we do not intend the toxicity prediction to vary based on the speaker's racial background, for example. If a model exhibits such variability, we call the phenomenon as *unintended bias*. In the rest of the paper, we use 'bias' to refer to unintended bias in toxic speech detection.

A simple bias mapping pipeline for training a generic NLP model is depicted in Figure 4.9. In the case of toxicity detection, biased sampling is performed to gain a higher percentage of toxic samples. Consequently, in broader NLP modelling, it occurs due to a skewed ratio in terms of quality and quantity of digital footprints of specific topics (such as content in support of LGBTQ vs against them). Sometimes, the training of NLP models can be unsupervised; in such cases, the annotation bias in the pipeline shifts to the last stage when it is employed on downstream tasks that use gold-label datasets to establish performance. All the shortcomings of human annotation apply to various NLP tasks. Some are more static and objective, leading to less bias and disagreements. Meanwhile, others can be highly subjective and ephemeral. As a result of various sampling and annotation biases, we observe the prevalence of identity-based harm introduced due to spurious lexical correlations.

As such, prejudice against specific identity groups can be analysed separately based on the target of harm. Inspired by existing approaches (Blodgett *et al.*, 2020; Sun *et al.*, 2019*b*; Shah *et al.*, 2020; Kumar *et al.*, 2021), we develop the following taxonomy based on the – (i) sources and (ii) targets of harm. Biases studied in this survey are not unique to the task of toxicity detection; instead, they are an extension of the biases in NLP applications. The taxonomy we develop allows us to approach the proposed methodologies from an applied perspective.

Based on *sources* of harm. We take inspiration from Suresh and Guttag (2021) to categorise based on the source of downstream harms during the data collection process. The process consists of selecting a *population*, selecting and measuring *features*, and *labels* to use. We study categories of bias according to the transformations related to these steps, such as sampling, lexical, and annotation bias.

Based on *targets* of harm. The following three categories of bias in toxic speech are each dedicated to a target group of downstream harm (Figure 4.10) – (i) racial bias, (ii) gender bias, and (iii) psychographic bias like political affiliations (Huszár *et al.*, 2021). The study of biases based on psychographic attributes has yet to gain popularity.

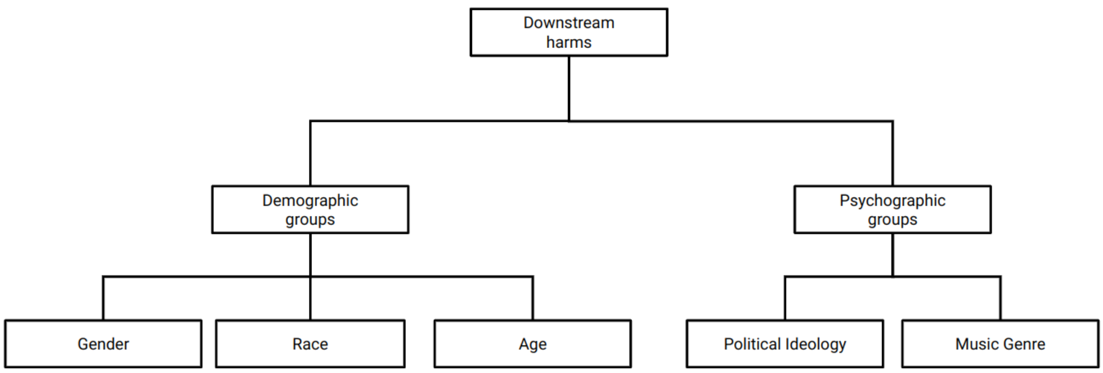


Figure 4.10: A taxonomy of bias based on the downstream harm. The harm can be inflicted through the deployment of a toxicity detection system.

4.4.3 Summary of findings

Tables 4.10 and 4.11 list the research and debiasing methods examined in this survey. Building on them, we summarise the major findings below:

- **Categories of bias.** The bias categories based on the *sources* of harm are not necessarily exclusive. Lexical bias can be a consequence of sampling bias, annotation

Table 4.10: A list of popular research studies is discussed in this survey. 'Analysis' indicates if the work only establishes the presence of bias. 'Mitigation' indicates if debiasing techniques are proposed. Note that this table is a representative sample of the literature surveyed here and is not a comprehensive set.

Bias	**Analysis**	**Mitigation**
Sampling	(Wiegand *et al.*, 2019; Ousidhoum *et al.*, 2020; Arango *et al.*, 2019; Razo and Kübler, 2020)	(Badjatiya *et al.*, 2019; Dixon *et al.*, 2018)
Lexical	(Wiegand *et al.*, 2019)	(Zhou *et al.*, 2021*a*; Badjatiya *et al.*, 2019; Dixon *et al.*, 2018; Kennedy *et al.*, 2020)
Annotation	(Waseem, 2016; Mozafari *et al.*, 2020*b*; Al Kuwatly *et al.*, 2020; Wich *et al.*, 2020)	
Racial	(Davidson *et al.*, 2019; Huang *et al.*, 2020; Davidson and Bhattacharya, 2020; Mozafari *et al.*, 2020*b*; Sap *et al.*, 2019)	(Xia *et al.*, 2020; Zhou *et al.*, 2021*a*; Sap *et al.*, 2019; Wiegand *et al.*, 2019; Park *et al.*, 2018)
Gender	(Huang *et al.*, 2020)	(Park *et al.*, 2018; Nozza *et al.*, 2019)

Table 4.11: A summary of debiasing methods employed for different types of unintended biases as discussed in this survey.

Bias	**Debiasing Method**
Sampling	Topic proximity (Van Der Wal *et al.*, 2022), Data mixing (augmentation) (Arango *et al.*, 2019)
Annotation	Disagreement deconvolution (Gordon *et al.*, 2021)
Lexical	Length sensitive sampling (Dixon *et al.*, 2018), Knowledge-based term generalisation (Badjatiya *et al.*, 2019), Data filtering (Zhou *et al.*, 2021*a*), Regularization (Kennedy *et al.*, 2020), Multi-task learning (Vaidya *et al.*, 2020), Ensemble modelling (Zhou *et al.*, 2021*a*)
Racial	Annotation priming (Sap *et al.*, 2019), Label correction (Zhou *et al.*, 2021*a*), Regularization (Schuster *et al.*, 2019), Adversarial training (Xia *et al.*, 2020), Ensemble modelling (Zhou *et al.*, 2021*a*)
Gender	Length sensitive sampling (Nozza *et al.*, 2019), Debiased LM (Bolukbasi *et al.*, 2016), Gender Swapping (Zhao *et al.*, 2018), Transfer learning (Park *et al.*, 2018)

bias, or both. Similarly, any category of bias based on downstream harms can be an artefact of all three categories based on the source of harm.

- **Sampling bias.** Our observations collectively suggest that the text source and the topics used for sampling have a more significant influence on the bias characteristics of the dataset than the sampling strategy. Additionally, OSNs have varying tolerance towards toxic speech (Munn, 2020). The difference in platform-specific policies can further affect the bias characteristics of the sampled dataset.
- **Lexical bias.** There should be a balance of sensitive words and phrases of explicit abuse, identity terms (especially those of minority groups), and topic words across the labels to avoid fabricating any spurious lexical correlations with the labels. These correlations can disrupt the model's ability to capture the context of such terms. A common point of failure of models trained on such data is the conflation of identity disclosures (e.g., 'I am gay') with identity attacks (e.g., 'I hate all gays'), further promoting disparity amongst social groups.

- **Annotation bias.** The current literature for annotating harmful content is spread on the spectrum from strict to loosely defined guidelines. Consequently, we observe that the annotation agreements also vary, with no fixed benchmark for toxic content labelling. The observations highlight the importance of examining the sensitivity of the annotators towards socio-cultural factors such as dialects and race. In order to effectively study annotators' behaviour across attributes, it is essential to have a fair number of participants, notwithstanding the fact that human annotations are resource and time-intensive.
- **Racial bias.** Every mitigation strategy suggested to reduce racial bias in existing models assumed that both Black-oriented and White-oriented samples follow the same conditional probability $p(Y|X)$. This assumption is flawed because using specific terms is socially acceptable for a Black person, while it can be unacceptable for someone else. Though existing resources (Blodgett *et al.*, 2016; Preoţiuc-Pietro and Ungar, 2018; Huang *et al.*, 2020) have aided the investigation of racial bias in toxic speech, none of them have access to the ground-truth labels for the dialect identities (Blodgett *et al.*, 2016). For example, it is possible that the accent detection LM spuriously correlates the presence of terms like '*n*gga*' and '*b*tch*' with the tweet being Black-oriented (Davidson *et al.*, 2019). Such predictions can magnify the racial bias during toxicity detection. Moreover, the assumption that only certain social groups employ specific terms is flawed in today's era. While on the one hand, it leads to broader adoption and acceptance of cultural terms, on the other hand, such terms are employed by extremist groups and trolls for misappropriation and mocking other cultures. Without knowing the background and intent of the online users, it is difficult to pinpoint what they wish to achieve when adopting non-native terms and dialects. The lack of awareness about dialects and their cultural importance among NLP researchers can itself be a source of bias.

4.4.4 Analogy from physical systems

We extend the interpretation that "*energy can neither be created nor destroyed but only transformed from one form to another*" to bias as a physical entity. One can say that as long as humans are bombarded with more information than they can process, some form of cognitive biases will keep existing as per the social structures of the zeitgeist. It also means that despite our best efforts, our interactions (online or offline) will be riddled with biases and stereotyping. Just like the "*system at rest remains at rest unless acted upon by an external force,*" to consciously and effectively overcome biases, we need to supply our NLP pipelines with external impetus in the form of debiasing frameworks. However, one-time mitigation and evaluation of bias, tested in the controlled setting, which is the current norm, will unfortunately not be an accurate (or even valid) indicator of bias reduction as the system evolves. The principle that the "*entropy of a closed system never decreases*" renders the one-time bias mitigation efforts moot. Translating entropy to the close-minded biases we harbour, our biases, once seeded, will not automatically disappear. Instead, bias, as an entity, is expected to be reinforced and strengthened with time. i.e., years of systematic discrimination and marginalisation leading to increased instances of hate speech (offline and online) and hate crime. Thus, we need a continuous probing setup (a regular supply of external energy) to keep the biases in the NLP pipeline, especially within toxicity detection systems, in check. We can only ensure their mitigation in an ever-evolving system of human interactions by evaluating biases at every step of the workflow and analysing them at regular intervals.

4.4.5 Limitations and future work

Despite covering multiple forms of bias evaluation and mitigation techniques, our work and the existing literature have limitations. Some of these challenges point to the general area of toxicity detection modelling and have been touched upon in our previous work (Chakraborty and Masud, 2022). This section discusses the limitations and possible solutions.

- **Cognizance towards side-effects.** Owing to resource constraints, the study of bias and its mitigation has focused on reducing only one bias at a time. Researchers often fail to acknowledge intersectionality while evaluating their proposed mitigation methods (Appendix F). Analysing intersectional bias is an open direction (Kim *et al.*, 2020). Most of the surveyed papers demonstrated bias mitigation as a single-step solution (Dixon *et al.*, 2018; Mozafari *et al.*, 2020*b*). However, it is essential to be "bias-aware" throughout the learning pipeline (Park *et al.*, 2018). A good reference has been formulated by Suresh and Guttag (2021). They formalise the complete pipeline as a sequence of data transformations and define the potential sources of harm.
- **Dynamic nature of language.** Existing literature in the area of gender debiasing in NLP, as well as in toxicity detection, has evaluated gender as binary (male vs female). As one observes in the existing analysis of annotation and lexical biases, annotators' lack of awareness of gender fluidity can lead to inconsistent labels. Extending from the previous point, the case of word reclamation is a part of the discussion around the use of static hate lexicons, offensive dictionaries, and static knowledge graphs, which cannot account for the evolving language and the evolving social-cultural aspects (Steels, 2016). Recently, Qian *et al.* (2021) have proposed a prototypical learned model for hate speech classification that aims to capture the evolving hateful content as it develops. Such models need to be extended to debiasing methods as well.

4.5 Chapter conclusion

As social media users become more adept at navigating the platform, they will employ writing techniques that bypass hate speech detection. The circumventing includes simple techniques like modified spellings or emoji substitution (Kirk *et al.*, 2022), references to implied stereotypes, and even humorous decoys. This reinforces the need for a closer examination of the concept of implicitness in toxicity analysis, which is introduced in this chapter. While `PolAt` gives a glimpse of political attacks in code-mixed Hinglish, `FiADD` presents a distance-metric-based PLM finetuning for implicit hate classification in English.

One of the key drawbacks of the contextual signal modelling from the last chapter was the overdependency on platform-specific signals. This is addressed and explored in this chapter in a two-fold manner. First of all, keeping true to the essence of contextual knowledge, this chapter has introduced an additional set of exogenous signals (as they are not platform-dependent) that are obtained as supplementary free text annotations. These annotations provide the required contextual information in the form of meaningful hints to decoding implicit hate by making it less implicit. Through successful modelling of this signal, we remove the need for APIs and web scrapers. Secondly, we also observe that the `FiADD` setup operates in a fashion where the contextual signal is available for training but not available at test time, which mimics the real-world scenarios better.

Our manual annotations extending the `ImpGab` and `AbuseEval` datasets get indirectly validated by the latent space analysis performed under `FiADD`. In subsequent parts of the thesis, we will revisit these signals and their integration with content moderation tools. It is interesting to note at this point that while free text annotations bring a more nuanced and 1-1 mapped context, they require manual effort to annotate. Balancing the human endeavour while maintaining the signal-to-noise in terms of contextual information will be a key towards building better content moderation tools.

Black box nature of toxicity detection and mitigation. Toxicity classification (via retweets, fine-grained labelling, or political attacks), as covered in the first two chapters of the thesis, forms the backbone of the content moderation pipeline. The first step is to filter and sort potentially harmful content for moderators to review. However, ML-based classifiers, whether feature-engineered or PLM finetuned, are a black box. It is hard to convey to both moderators and end users why a specific text is labelled as hateful by the classifier. Why a post or a user has been flagged as offensive is vital to convey why the flagging occurred. On one hand, we employ contextual signals to improve performance and reduce the model-human knowledge gap. On the other hand, post-modelling interpretability tools still explain the output from a technical/performance metric perspective, which is hard to translate into layperson's terms. Keeping the biases in check can lead to better generalisation. Another exciting area of research can be employing weakly supervised or unsupervised methods of data augmentation (Sarwar and Murdock, 2022) and domain adaptation (Ludwig *et al.*, 2022) to reduce the need for annotations and improve the generalisability of toxicity detection models. Setting up a feedback infrastructure, wherever possible, allows a collection of data that better represents the target population.

With the focus on LMs shifting from classification-based PLMs to generative PLMs and instruction-tuned LLMs, a closer examination of the generative power of PLMs and LLMs in the content moderation pipeline is a step in the right direction. The free text-based labelling allows for the modelling output to be consumed in a more manageable/human-readable form, even though the internal working of the model is still a black box.

Part III

User Centric Tooling

CHAPTER 5

Generative Tools for Better Human Interaction

"As a society, we must combat polarisation and take concerted action when intolerance raises its head."
- Sigrid Kaag; Kristallnacht Commemoration

5.1 Chapter introduction

In the first part of the thesis, we empirically establish that contextualised systems, whether feature-engineered or PLM-based, are efficient at classifying hatefulness. However, their outputs are not immediately discernible by humans. Removing content from a platform without a clear explanation/justification leaves both content moderators and the end user confused. In the absence of concrete explanations and nudging, there is less room for corrective behaviour in the future. It is crucial to acknowledge that the interpretability of classifiers is a broader concern within the machine-learning community and is an active area of research (Zhang *et al.*, 2021). In terms of content moderation, we use a slightly different approach. *Instead of exploiting the discriminative capabilities of PLMs, we turn to their generative capabilities instead.* The generative nature of PLMs allows us to develop tools where humans can directly consume the model's output in natural language. As a by-product of readability and user-friendly production, the aim is to encourage better decision-making. In this chapter, we explore how to empower content moderators (humans or machines) to flag implicit content better. In the next chapter, we address proactive nudging of users to post less explicit content.

Moving towards generative PLMs necessitates exploring newer contextual signals. The de facto technique has been to augment the PLMs with knowledge graph (KG) tuples (Sridhar and Yang, 2022; Chang *et al.*, 2020; Lin, 2022) that enhance the model's reception of world knowledge and trigger more attention towards the entities in the tuples. Despite the reported improvements in performance metrics, we hypothesise that the KG tuples do not account for the multi-hop/indirect nature of implicit hate. Working with two publicly available datasets and two KGs, we observe the same. Pithily speaking, replacing the top-k most relevant tuples with either the bottom-k least relevant tuples or random-k tuples does not necessarily cause the relative performance to languish. Looking beyond KG-infusion, we investigate the infusion of "toxicity attributes." Consequently, simpler models incorporating external toxicity signals outperform KG-infused models. Against the KG-based setup, we observe a comparable performance with `Tox-BART` (`toxicity-infused-BART`) for `SBIC` (`LatentHatred`) datasets with a performance variation of +0.44 (+0.49), +1.83 (-1.56), and -4.59 (+0.77) in BLEU, ROUGE-L, and BERTScore. Further human evaluation and error analysis reveal that our proposed setup produces more precise and explicit explanations than zero-shot GPT-3.5, highlighting the intricate nature of uncovering implicitness.

5.2 Auditing knowledge graphs for implicit explanations

5.2.1 Motivation

Research gap. Elucidating the underlying implied hate is a non-trivial task. Here, finetuning is often augmented with KG tuples (in the form of entity$_1$-relation-entity$_2$) (Sridhar and Yang, 2022; Chang *et al.*, 2020; Lin, 2022; ElSherief *et al.*, 2021) at the input level to reduce the information gap. To reduce the signal-to-noise ratio for a given input query, practitioners filter only the top-most relevant tuples. Apart from reporting changes and improvements in standard performance metrics, there is little to no ablation around how the underlying KG, the tuple retrieval process, and task specificity impact the quality of the extracted KG tuples, which in turn affects the performance metrics. This investigation is all the more critical within hate speech literature. Recently, it has been observed that Wikipedia entities, as sources of contextual knowledge for implicit hate classification, do not help at a fine-grained level (Lin, 2022). We hypothesise similar behaviour with KG infusion.

Research question. We aim to empirically examine:

RQ1: What (if any) is the relation between the "quality" of knowledge tuples and the PLM-generated explanations for implicit/stereotype statements?

Contribution summary. We believe that the process of obtaining KG tuples, being task-agonistic, may not lend itself to task specificity. We study this phenomenon in the light of PLM-generated explanations for implicit hate. However, directly establishing the causal relation between the quality of KG tuples and the generated output from PLMs like BART is intractable. Instead, we postulate that if adding top-k KG helps improve a model's generation capabilities, then the generations should deteriorate when the top-k is corrupted. Working with two publicly available implicit explanation datasets – `SBIC` (Sap *et al.*, 2020) and `LatentHatred` (ElSherief *et al.*, 2021), and two diverse KGs – `ConceptNet` (Speer and Havasi, 2012) and `StereoKG` (Deshpande *et al.*, 2022), we observe that if less relevant counterparts replace the top-k tuples, the performance difference is insignificant for the task of implicit explanation generation. To assert this anomaly, we perform a two-part error analysis to corroborate our hypothesis. Our findings are a starting point in addressing the gap (Yadav *et al.*, 2024).

5.2.2 Experimental setup

Datasets for evaluation. We employ two standard implicit explanation datasets in English. Apart from `LatentHatred` (ElSherief *et al.*, 2021), we also look at `SBIC` (Sap *et al.*, 2020). Both are a parallel corpus of an input post obtained from the web containing implicit hate ($\mathcal{X}$) and the corresponding stereotype explanation ($\mathcal{Y}$) obtained via human annotations. A single post in `SBIC` can have multiple free-text annotations. For `LatentHatred`, every post has a single annotation, along with a class label of explicit, implicit, or neutral. Here, the explanations are only available for the implicit hate samples. The dataset statistics are listed in Table 5.1. Hateful posts for `SBIC` are sourced in equal parts from Reddit, Twitter, and ExtremeHate Forums (Gab, Stormfront, BannedReddit). Meanwhile, `LatentHatred` is solely curated from Twitter.

Data processing. We also follow the data preprocessing pipeline adopted from MIXGEN, where we replace NAN, URL, and unique tokens. We lowercase the samples. The respective datasets already mask sensitive user information. We do not perform any further masking.

Table 5.1: Dataset statistics for `SBIC` and `LatentHatred`. Here, 'post' is the input implicit statement, and 'implied' is the implied stereotype. We report both features' average (standard deviation) token length (len).

Feature	SBIC		LatentHatred	
	Train	**Test**	**Train**	**Test**
# Samples	35933	4705	5722	636
Post len.	107.0 (63.3)	107.0 (65.6)	94.0 (40.0)	31.0 (11.7)
Implied len.	16.0 (15.3)	19.0 (14.5)	96.0 (43.8)	31.0 (11.7)

Table 5.2: Comparison of the KGs involved in our investigation.

KG/Property	**ConceptNet**	**StereoKG**
Size (# tuples)	$\sim$34M	$\sim$4k
Curated from	Wikipedia	Reddit (offensive subreddits)
Type of tuples	World and common-sense knowledge	Religious and ethnic stereotypes
Top-k tuples via	Weighted TF-IDF	Cosine Similarity

External knowledge sources. To corroborate our findings, we employ one general-purpose and one task-specific KG. `ConceptNet` (Speer and Havasi, 2012) is a KG consisting of $\approx 34M$ tuples/assertions of world knowledge and common sense relations curated from Wikipedia. On the other hand, `StereoKG` (Deshpande *et al.*, 2022) is a nascent KG with $4k$ tuples capturing stereotypes from Twitter and Reddit. Given the intention of capturing stereotypes in social media posts, `StereoKG` is closest to being an ideal KG for our task. An overview of the KGs is provided in Table 5.2

Problem statement. We follow the setup provided in MIXGEN (Sridhar and Yang, 2022) - a Knowledge Graph (KG) infused BART-based model for implicit explanations. MIXGEN serves as the baseline work considered in this chapter. Borrowing from MIXGEN, we concatenate the input ($\mathcal{X}$) with k KG tuples ($t_1, t_2, \ldots, t_k$) as $\tilde{\mathcal{X}}$ = $\{\mathcal{X}$, $[SEP]$, t_1, $[SEP]$, t_2, $[SEP]$, ..., $t_k\}$, where $[SEP]$ is the separator token. $\tilde{\mathcal{X}}$ is then input to BART. The aim is to reduce CE over the predicted generations $\tilde{\mathcal{Y}}$, where $\tilde{\mathcal{Y}} = PLM(\tilde{\mathcal{X}})$.

PLM. We use the 'bart-base' (Lewis *et al.*, 2020) as our PLM for examination, again following MIXGEN. As a control setup, we compare the KG-infused BART with vanilla BART, where no tuples are infused at the input level, i.e., $\tilde{\mathcal{Y}} = PLM(\mathcal{X})$. For semantic embedding comparison during KG retrieval, we employ SBERT's `all-MiniLM-L6-v2` (Reimers and Gurevych, 2019).

Evaluation metrics. We employ standard evaluation metrics in NLP – BLEU (Bilingual evaluation understudy) (Papineni *et al.*, 2002), ROUGE-L (Recall-Oriented Understudy for Gisting Evaluation) (Lin, 2004), and BERTScore (Zhang* *et al.*, 2020) to measure the syntactic, linguistic, and semantic similarities between the generations and gold labels.

5.2.3 Filtering k-relevant tuples

For ConceptNet, we follow the retrieval method proposed by Chang *et al.* (2020) and Sridhar and Yang (2022). In this setup (*Algorithm 1*), we first obtain the query terms (q) from the input post's lemmatised noun, verb, and adjective keywords. We then extract from ConceptNet all the 1-hop English tuples for each term. We also calculate the IDF score for each query term, idf_q. The top-k and bottom-k tuples are obtained by sorting the extracted relations based on relevance scores $W_{rel} \times idf_q$, as each relation in ConceptNet has a *relation-weight*, W_{rel}. For random-k, we randomly pick k tuples from the extracted set. The pseudo-code is listed in Algorithm 1.

Algorithm 1 Knowledge Tuples Extraction for ConceptNet

Require: Input: A post $\mathcal{X}$. Output: $KG_h = \{(r_i, t_i, score_i) \mid 0 \leq i \leq N_i\}$
1: $x_1, x_2, x_3, ... \leftarrow$ word level tokenised $\mathcal{X}$
2: $h \subseteq x_1, x_2, x_3, ... \leftarrow$ extract adjectives, nouns, and verbs from each post, i.e, $query\_tokens$
3: $KG_h = \{(r_i, t_i, rel_i) \leftarrow r_i \in h$, and rel_i is the relation weight between r_i and t_i
4: $idf\_scores_i \leftarrow$ TF-IDF scores for $r_i \in h$
5: Sort tuples in KG_h in terms of $idf\_scores_h \cdot score_i$ **return** KG_h where $|KG_h| = N$

For StereoKG, we utilise a semantic similarity-based metric (*Algorithm 2*). We first employ SBERT to pre-calculate the sentence embeddings over all the linearised tuples. The linearised tuples are already provided along with the triplets in StereoKG. We then use the cosine-similarity scores between tuples and input samples to get the top and bottom-k tuples. The pseudo-code is outlined in Algorithm 2.

The same algorithm cannot be applied to both KGs due to the skewness in KG size. Employing *Algorithm 1* for StereoKG returns a low (and zero in most instances) number of tuples per input sample. Meanwhile, employing *Algorithm 2* for querying on ConceptNet is not computationally feasible as it amounts to performing cosine similarity in the order of millions. Further, following the experimental setup from MIXGEN, we also set $k = 20$.

Algorithm 2 Knowledge Tuples Extraction for StereoKG

Require: Input: A post $\mathcal{X}$. Output: $KG_h = \{(r_i, t_i, score_i) \mid 0 \leq i \leq N_i\}$
1: $lin\_KG \leftarrow$ Linearised tuples from StereoKGvia model $\mathcal{Q}$
2: $emb\_vec \leftarrow$ embedding $\mathcal{X}$ via model $\mathcal{Q}$
3: $score_i \leftarrow cosine\_sim(emb\_vec, lin\_KG)$
4: Sort tuples in terms of $score_i$. **return** KG_h where $|KG_h| = N$

5.2.4 Results

Table 5.3 shows that compared to vanilla BART, LatentHatred's performance improves under all KG infusions. Meanwhile, due to KG infusion, SBIC is more varied and even registers a drop in BLEU and ROUGE-L. More interestingly, we have counter-intuitive results comparing the three top/bottom/random-k configurations. In 3/4 combinations, *the performance difference is visibly insignificant (and in some instances*

Table 5.3: `SBIC` and `LatentHatred`'s performance variation across `ConceptNet` and `StereoKG` in terms of – B: max-BLEU; R: ROUGE-L F1; BS: BERTScore F1, where $k = 20$ is the best-performing hyperparameter of MIXGEN (Sridhar and Yang, 2022).

Method	ConceptNet						StereoKG					
	SBIC			LatentHatred			SBIC			LatentHatred		
	B	**¯R**	**BS**	**B**	**¯R**	**BS**	**B**	**¯R**	**BS**	**B**	**¯R**	**BS**
BART Baseline	72.17	70.83	78.05	38.38	17.65	90.37	72.17	70.83	78.05	38.38	17.65	90.37
Top-k	68.41	66.4	80.37	47.23	36.26	92.12	63.57	61.30	76.39	46.39	35.37	92.03
Bottom-k	68.97	66.80	80.95	47.40	35.90	92.15	60.31	58.09	73.44	46.92	35.94	92.04
Random-k	69.69	67.47	81.63	48.34	37.18	92.31	60.80	58.45	73.87	47.27	36.12	92.07

Table 5.4: Pair-wise effect size and p-test on (B: max-BLEU; R: ROUGE-L F1; BS: BERTScore F1) when comparing the column-wise control group with the row-wise treatment group for `LatentHatred` on BART with `ConceptNet` and `StereoKG` respectively, with $k = 20$. * ($p \leq 0.05$) and **($p \leq 0.001$) indicate whether the difference is significant.

KG		Base	T	B
C	**T**	2.19**, 2.56**, 1.74**		
	B	2.06**, 2.23**, 1.89**	0.25, -0.01, -0.15	
	R	2.02**, 2.18**, 1.46**	0.28, -0.21, -0.28	-0.00, -0.22, -0.17
S	**T**	2.21**, 2.00**, 1.33**		
	B	2.12**, 2.06**, 1.71**	0.37, 0.24, 0.33¯	
	R	2.24**, 2.49**, 1.42**	0.25, 0.08, -0.16	-0.09, -0.13, -0.40*

even increases) if we replace top-k with bottom-k or random-k tuples. The above results also highlight that the influence of KG on a dataset varies on a case-by-case basis.

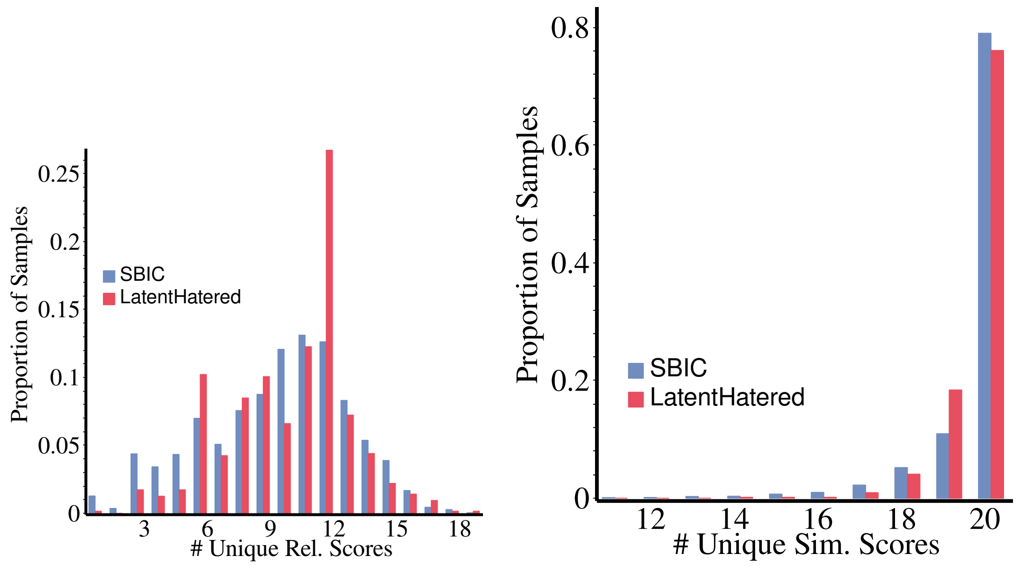
Given the higher improvement in performance for `LatentHatred`, from no-KG to KG infusion, we also report the paired t-test and each pair's effect size for `LatentHatred`. Based on Table 5.4, we see that going from standalone/vanilla BART to KG infusion (top, bottom, or random) leads to a significant increase in performance, as corroborated by a considerable effect size (≥ 1) and $p \leq 0.01$ in all metrics for the "Base" column in both `StereoKG` and `ConceptNet`. On the other hand, among top-k, bottom-k, and random-k, in Table 5.3, the insignificant ($p > 0.01$) effect size indicates that the variation is considerably negligible. It also implies that replacing one with the other will not significantly alter the performance among top, bottom, and random-k.

> Significance testing of Table 5.4 corroborates the results in Table 5.3 that there appears to be no noticeable deviation in expected behaviour when replacing most relevant tuples with bottom or random-k tuples.

5.2.5 Auditing relevance scores

Intrigued by the observations in Section 5.2.4, we look into the values of retrieval scores for the case of top-k tuples. The range for retrieval scores termed as relevance and similarity scores, respectively, for `ConceptNet` and `StereoKG` is $[0, \inf)$ and $[0, 1]$. In Figure 5.1, we record the number of unique scores obtained for the test set. For $k = 20$, one would expect the uniqueness to be right-skewed, which is partially valid for `StereoKG` but not for `ConceptNet` where there are fewer samples with $>= 16$ unique scores and zero samples with all unique scores. The similarity metric being limited to $0 - 1$ for `StereoKG` but being open-ended ≥ 0 for the `ConceptNet`, one would expect the latter to generate more variation in scores.

Moreover, Figure 5.2 shows that patterns of scores per KG are similar for respective hate datasets, proportional to the number of test samples in each. The majority of relevance scores w.r.t `ConceptNet` are ≤ 1 and only 3.5% (1.5%) of samples of `SBIC` (`LatentHatred`) garner scores $>= 5$ for at least one of the tuples. The similarity scores for `StereoKG` are also on the lower end, with the majority covered in the range $0.3 - 0.5$. *These observations indicate low-quality tuples getting filtered in top-k.* Based on top-k retrieval scores, bottom-k and random-k should be equally low-quality.



(a) ConceptNet

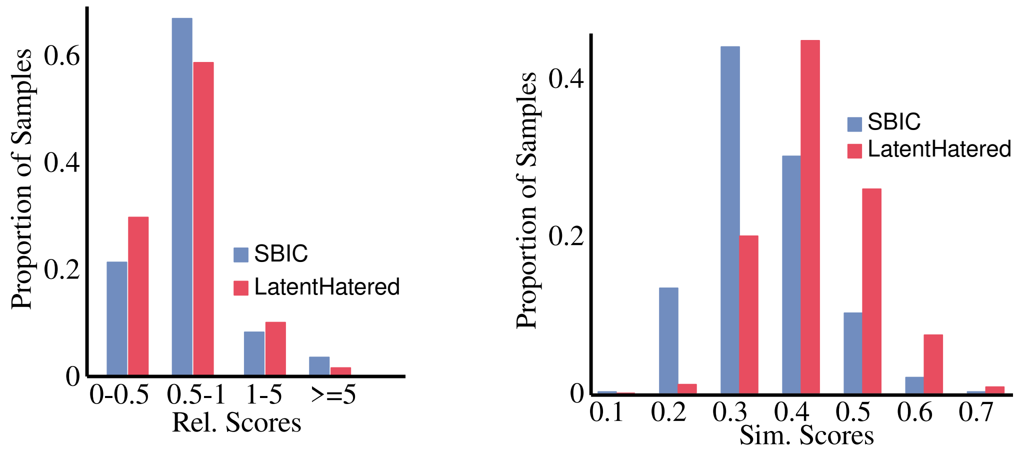
(b) StereoKG

Figure 5.1: Analysis of top-k KG tuples of `SBIC` and `LatentHatred` at $k = 20$, capturing the spread of uniqueness in scores obtained per sample, respectively, for `ConceptNet` and `StereoKG`. The ith index on the x-axis is the number of unique scores present in the top-k samples.

5.2.6 Manual auditing of KG tuples

Building upon the observations of the previous section, to concretely ascertain the low quality of KG tuples, we manually inspect randomly selected 20 samples from `SBIC` and `LatentHatred` each.

Manual assessment. The corresponding top-k tuples are extracted w.r.t `ConceptNet` and `StereoKG`. Two expert annotators (detailed below) score each pair (input, top-k set) per KG. The manual labelling captures two components, 'task-domain relevance' and 'general-domain relevance,' scored separately on a 5-point Likert scale. Task-domain relevance determines how effectively the retrieved tuples can explain implied stereotypes. A general-domain relevance determines if the tuples capture diverse concepts enlisted in the sentence from a commonsense/world-sense understanding. Besides providing scores, the annotators can offer any additional comments about an outlier they observe.



(a) ConceptNet (b) StereoKG

Figure 5.2: Analysis of top-k ($k = 20$) KG tuples for `SBIC` and `LatentHatred`, capturing the spread of raw score values for (a) `ConceptNet` and (b) `StereoKG` respectively. Here, the x-axis represents the score value as either binned (for `ConceptNet`) or rounded to the nearest 1st decimal (for `StereoKG`). The bins range from [start, end) except for the last bin.

Annotator demographic. We take help from 2 expert annotators who volunteer ≈ 35 minutes each and score 20 samples and their $top - k = 20$ KG tuples. The annotators, one male (24 years) and one female (29 years), are knowledgeable about natural language processing and social computing. Additionally, both adequately understand how KGs are constructed and employed in NLP.

Observations. At the end of the annotation process, we obtain 320 manual scores summarised in Table 5.5. Table 5.5 lists the average (per annotator) scores and the inter-annotator cosine scores. We also observe a higher alignment of tuples in `LatentHatred`, which explains the improvement in performance registered by this dataset under KG infusion (Table 5.3). As a toxicity-specific KG, `StereoKG` seems to provide comparatively better tuples than `ConceptNet`, yet both end up with abysmal relevance scores garnered by both annotators. *Our manual inspection strongly corroborates that the quality of tuples is not informative/specific enough for our task.*

> Our extensive evaluations encompassing standard evaluation metrics, significance testing, and automatic and manual auditing of KG tuples reveal that the defacto tuple retrieval filtering is not contextually sufficient to explain implicit hate. The absence of explicit hate or indirect mention of the target means that extracted entities may not relate to hateful connotations.

5.2.7 Broader research implications

The KG infusions fall short of eliciting latent cognitive capabilities for social reasoning/subjective tasks such as implicit hate. Similar issues in implicit hate detection tasks have been observed via automated evaluations (Lin, 2022). However, ours is one of the initial works to look into this issue extensively. We suspect such behaviour will occur in other NLP tasks, such as explanations of sarcasm (Kumar *et al.*, 2023). There is a need

Table 5.5: Task (T_r) and general domain (G_r) relevance scores by annotators A_1 and A_2 on 20 random SBIC and LatentHatred samples. We report the mean (standard deviation) scores. Cosine similarity captures the IAA w.r.t ConceptNet (C) and StereoKG (S).

D	KG	A_1		A_2		Cosine Sim.	
		T_r	G_r	T_r	G_r	T_r	G_r
SBIC	C	0.24 (± 0.44)	0.29 (± 0.46)	0.05 (± 0.50)	0.95 (± 0.70)	0.47	0.56
	S	0.43 (± 51)	0.43 (± 51)	0.19 (± 0.03)	0.52 (± 0.36)	0.52	0.68
LatentHatred	C	0.3 (± 0.57)	0.65 (± 0.75)	0.2 (± 0.41)	0.4 (± 0.60)	0.71	0.73
	S	2.35 (± 67)	1.55 (± 0.89)	1.15 (± 0.59)	1.35 (± 0.67)	0.89	0.79

for domain-specific KG retrieval and ranking methods for KG tuples. Regarding augmenting KGs, research in this area will benefit from efficient task-specific and multi-hop retrieval functions to enhance the quality of top-k tuples. Parallelly, there must be an active discussion on "how LMs learn the association between external and pretrained features?"

5.2.8 Limitations and future work

- **Variability in PLM and tasks.** While the current set of experiments focuses on a single PLM for the task of hate speech, a broader analysis of this both within hate speech and within NLP is needed. While the size of the PLM does not impact the quality of the tuple extracted, it will be interesting to observe how the concatenation of tuples impacts the performance of PLMs of larger size.
- **Flattening of non-sequential entities.** Our work, following the standard practice, infuses the tuples in a sequential/serialised manner. Given that implicitness can be expressed in multiple forms and encompasses multi-hop connotations, the work also calls for better infusion of non-sequential information, such as knowledge graphs, into seq-2-seq LLMs (Besta *et al.*, 2024).
- **Contextual signals beyond knowledge entities.** Our research with KG tuples highlights the gap in contextual understanding of general knowledge frameworks for hate speech-related tasks. Now that the quality of KG does not have a significant impact on the quality of generated explanations, it is imperative to explore other signals.

5.3 Toxicity attributes for explaining implicit hate

5.3.1 Motivation

Research gap. Implicit hate uses circumlocution and stereotyping to mask the hate (Gao *et al.*, 2017*a*), which content moderation systems (human or computer-aided) sometimes fail to understand. We observe this even with sophisticated systems like ChatGPT (GPT-3.5). The line of work explaining toxic text (Cao *et al.*, 2022; Balkir *et al.*, 2022) or generating implications for stereotypes (Sridhar and Yang, 2022; Sap *et al.*, 2020; ElSherief *et al.*, 2021) is nascent and primarily employs LLMs (Zhou *et al.*, 2023; Mun *et al.*, 2023; Zhang *et al.*, 2023). Meanwhile, post hoc attention scoring and rationale-based training techniques fail to detect implicit spans (Mathew *et al.*, 2021;

Masud *et al.*, 2022). While KG infusion has been the go-to practice, in the previous Section 5.2, we record their lack of task specificity. This leaves a gap in both determining newer contextual signals to employ and how they should be infused.

Research questions. Revisiting the `SBIC` and `LatentHatred` datasets, in this work, we primarily focus on the following issues:

- **RQ1:** What alternate signals can be leveraged to explain implied stereotypes?
- **RQ2:** How and when in the finetuning pipeline should the explanatory signals be best incorporated for maximum performance gain?
- **RQ3:** Can implied explanations bring out the hidden toxicity to help content moderation?

Contribution summary. We investigate the infusion of "toxicity attributes" as contextual signals and introduce `Tox-BART`. These "toxicity attributes" (AlKhamissi *et al.*, 2022) can be defined as indicators outside the post text that convey the power dynamics (Zhou *et al.*, 2023), target groups (Sap *et al.*, 2020), insult-type (ElSherief *et al.*, 2021), or hate intensity (Masud *et al.*, 2022) of the post. Compared to the KG-based setup, `Tox-BART` observes a comparable performance for `SBIC` (`LatentHatred`) datasets with a performance variation of +0.44 (+0.49), +1.83 (-1.56), and -4.59 (+0.77) in BLEU, ROUGE-L, and BERTScore. We also look into how varying the quality of toxicity signals leads to the expected loss in performance (unlike in KG infusion), which is another indicator of the consistency of `Tox-BART`. Based on standard metrics, human evaluation, and error analysis, we observe that `Tox-BART` outperforms GPT-3.5 by producing more specific explanations. We conclude the discussion with a demo of how `Tox-BART` explanations can be employed for content moderation (Yadav *et al.*, 2024).

5.3.2 Toxicity attributes as contextual signals

We broadly classify toxicity attributes as – *in-dataset* or *in-domain*. In Figure 5.3, we provide an outline for how these signals can be augmented in explaining implicitness.

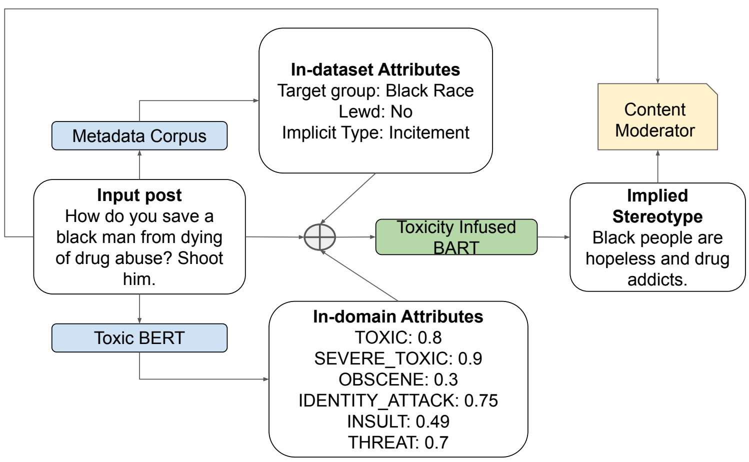


Figure 5.3: Workflow of our proposed setup utilising toxicity attributes (*in-dataset* and *in-domain*) for explaining implicit hate.

In-domain attributes. As the name suggests, these signals are external to the dataset but related to the "domain" of hate speech, conveying information about the harmfulness of the incoming posts. The KG tuples from `StereoKG` qualify as *in-domain* attributes. For our use case, the toxicity indicators are obtained from the Jigsaw dataset (Adams *et al.*, 2019). In Jigsaw ($\approx 2M$ samples), an input text j has multiple annotations, with each annotator giving a score between $0 - 1$ for labels $t_1, t_2, \ldots, t_6 \in$ {toxicity, severe toxicity, obscene, threat, insult, identity attack}.

In-dataset attributes. In layman's terms, they are akin to "metadata" providing complementary information. For example, if the post comes from Twitter, then the likes and reply count (Founta *et al.*, 2019) become engagement-based metadata features. Other times, this information can be completely unsupervised/unlabelled but still functional, like the user's ego network (Kulkarni *et al.*, 2023). We already explored unsupervised and pseudo-labelled metadata information in the form of exogenous signals in Chapter 3.1. In this section, we explore supervised, *a.k.a* annotated metadata. These annotations can be generated manually or by machine, but are usually available as 1-1 mapped auxiliary labels about the speaker, target, etc., sometimes available within the dataset itself, and can sometimes be generated based on the context (Zhou *et al.*, 2023). For example, both `SBIC` and `LatentHatred` have free text annotations for the target group. `SBIC` further has labels indicating whether the incoming posts are (a) *intentional*, (b) *lewd*, (c) *offensive*, (d) *targeting a group*, and (e) *uses in-group language*. In the case of `SBIC` the labels are independently binary. Meanwhile, `LatentHatred` has labels indicating the type of implicit hate from among – *grievance, incitement, inferiority, irony, stereotypical, threatening*, or *other*.

5.3.3 Design of `Tox-BART`

We first outline the in-domain ($\mathcal{P}$) and in-dataset ($\mathcal{A}$) attributes and then formulate multiple configurations of `Tox-BART`. `Tox-BART` is tuned on a pair of implicit input posts ($\mathcal{X}$) and implied explanations ($\mathcal{Y}$). We denote the BART encoder/decoder with $\mathcal{F}_\theta/\mathcal{G}_\theta$, with $d \in \mathbb{R}^{768}$ embedding dimension and θ trainable parameters. We experiment with multiple configurations and list the best configurations here. Additional configurations are discussed in Appendix G. Note that the design principles driving our research at this point are an easy-to-use system and deployable for the benefit of content moderators while being on par with KG-infused defacto systems in terms of performance metrics. Here, we finetune PLMs as they are openly available to begin with. Interestingly, we observe that simple fintuned configurations also provide the best performance.

Best configuration for in-domain attributes

Formally, given a regressor $\mathcal{R}_\phi$ with parameters ϕ, input j, and labels t, we minimize $\mathcal{L}_{MSE} = \frac{1}{n}\sum_{i=1}^{n}(\mathcal{R}_\phi(j_i) - t_i)^2$. The output of `ToxicBERT` is a vector $\mathcal{P} \in \mathbb{R}^{1*6}$, where each dimension represents the probability for the type of insult. We explore different configurations to infuse $\mathcal{P}$ with the incoming post $\mathcal{X}$ (Figure 5.4). Below, we expand on the best configurations. Appendix G outlines the rest.

Configuration 1 (C1). As the probability values (regression results) in isolation do not convey information about the feature they are implying a presence or absence of, we propose converting the values into their corresponding label tags. Apart from providing direct context about toxicity, employing text provides uniformity across train samples, as chances of text attributes co-occurring are higher than that of exact probability score

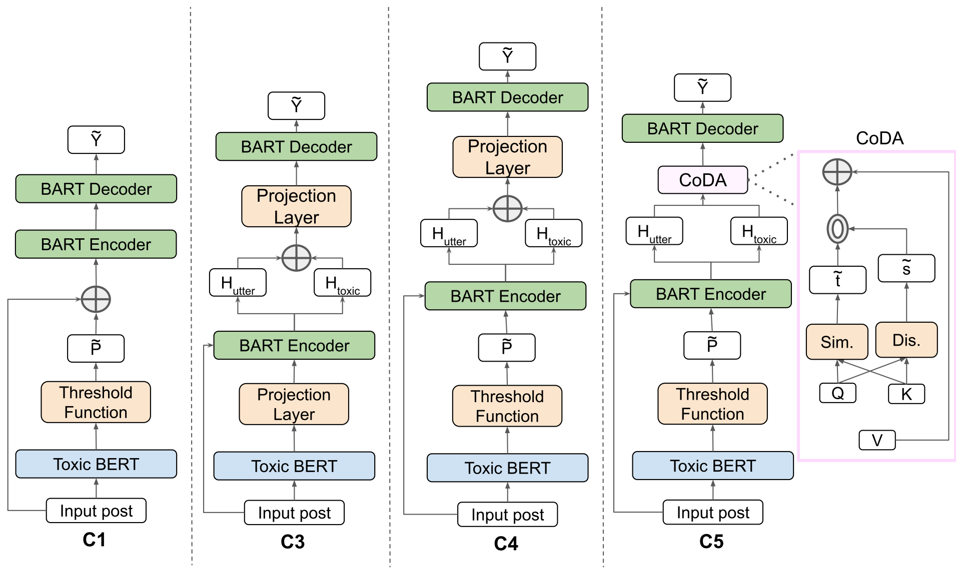


Figure 5.4: Configurations (C1, C3-C5) for incorporating *in-domain* attributes via the `ToxicBERT` regressor. BART encoded toxic attributes, and input representations are $\mathrm{H}_{\mathrm{toxic}}$ and $\mathrm{H}_{\mathrm{utter}}$, respectively. $\tilde{\mathcal{P}}$ is the modified toxic attributes vector, whereas $\tilde{Y}$ is the system generated explanations. The toxic-attributed BART is our proposed system - `Tox-BART`. In C5, Sim. (Dis.) captures query (Q), Key (k), and value (V) for the similarity (dissimilarity) matrices. $\tilde{t}$ and $\tilde{s}$ represent the tanh and sigmoid functions, respectively.

vectors. We convert the probability values into their corresponding toxicity tokens via a threshold parameter (λ). For instance, if p_i captures the probability score for the label "threat," then based on $p_i < \lambda$, its equivalent textual presentation will be a special token either <NOT_THREAT> or <THREAT>. The six toxicity tokens are then concatenated (using $[SEP]$) to the incoming posts ($\mathcal{X}$). Equation 5.1 outlines the setup, where Γ corresponds to the probability scores of the toxicity-token transformation function parametrised by λ.

$$\tilde{\mathcal{P}} = \Gamma(\lambda, \mathcal{P}), \quad \tilde{X} = [\mathcal{X}, \tilde{\mathcal{P}}]; \quad \tilde{\mathcal{Y}} = \mathcal{G}_\theta(\mathcal{F}_\theta(\tilde{X})) \tag{5.1}$$

Best configuration for in-dataset attributes

Configuration 2 (C2). For n *in-dataset* attributes $A = \{A_1, A_2 \ldots A_n\}$ for an input post, we first concatenate them using whitespace[1] ($\tilde{A} = [A_1[w]A_2.\ldots.[w]A_n]$) and then concatinate $\tilde{A}$ with input post as outlined in Equations 5.2.

$$\tilde{X} = [X, \tilde{A}]; \tilde{\mathcal{Y}} = \mathcal{G}_\theta(\mathcal{F}_\theta(\tilde{X})) \tag{5.2}$$

Overall loss

We aim to reduce CE loss over the predicted generations $\tilde{\mathcal{Y}}$ infused by toxicity attributes ($\mathcal{P}$ or $\mathcal{A}$) in `Tox-BART` based on Equation 5.3.

[1]We experiment with SEP token as well, but whitespace worked better here.

$$\ell(\Theta) = \ell_\theta^{CE}(\mathcal{Y}, \tilde{\mathcal{Y}}) \tag{5.3}$$

5.3.4 Experimental Setup

Note on datasets. We work with the same datasets (`LatentHatred` and `SBIC`) and preprocessing pipeline that we employ for KG assessment in Section 5.2.2.

Hyperparameters. We use the HuggingFace Transformers Library for our experiments, with 'bart-base' being our backbone PLM and 'bert-base' for the toxic attribute probability approximation task. To reiterate, the hidden state dimensions for both BART and BERT are 768. For inference, borrowing from Sridhar and Yang (2022), the length penalty is set to 5, and the number of beams for beam search is set to 10. The experiments are collectively performed over an NVIDIA RTX A5000 and A6000.

Evaluation metrics. Similar to the last experiment, we employ BLEU, ROUGE-L, and BERTScore for automated evaluation.

Baselines. We consider the following setups for comparison:

- **Finetuned vanilla BART.** We finetune without any external attribute.
- **Finetuned vanilla GPT-2.** We follow Sap *et al.* (2020)'s implementation of generating the stereotypical implication.
- **Finetuned MIXGEN.** Sridhar and Yang (2022) proposed three setups; however, we were able to reproduce only two of their modules, i.e., the *explicit knowledge* and *implicit knowledge* models. External knowledge stored within KGs, which PLMs can utilise, is termed explicit knowledge. In contrast, the knowledge stored inherently within the PLM is implicit knowledge. Subsequently, when combining explicit and implicit knowledge, MIXGEN utilises the top-k most relevant ConceptNet tuples and GPT-2 prompted outputs[2].
- **Zero-shot GPT-3.5.** We also present a comparison with the zero-shot generations of `gpt-3.5-turbo`. We employ the following prompt for generating implications *"What stereotype is propagated by this post: [POST]? Answer in simple words and keep the length less than 5-10 words."* We do not perform extensive prompt engineering for GPT-3.5. However, after the initial investigation, we add the phrase "answer in simple words and keep the length short" to reduce wordy[3] and non-contextual explanations like "People should not indulge in hateful content."

5.3.5 Results

To establish the efficacy of "toxicity attributes," we conduct extensive automatic evaluation comparing `Tox-BART` with KG and non-KG-based systems. Further, we show the robustness and sensitivity of `Tox-BART` via ablation.

[2]We observe a significant deviation in results reproduced for MIXGEN. Since we did not change or tune any hyperparameters from the original MIXGEN setup during training and inference, this discrepancy can arise from hardware or random seeding currently missing from MIXGEN.

[3]Based on dataset statistics in Table 5.1, the mean explanation length is ≈ 25 words.

`ToxicBERT`. The RMSE scores on train and validation set of D_{jigsaw} for `ToxicBERT` are enlisted in Table 5.6.

Table 5.6: RMSE for the best checkpoint of `ToxicBERT`.

Split	**Loss**
Train	0.0592
Validation	0.06887

Automated evaluation. Interestingly, for `SBIC` the infusion of non-PLM external signal, either KG or toxicity, leads to a drop in its performance compared to vanilla BART. On the other hand, the `LatentHatred` dataset proves more difficult with fewer samples to train on; however, infusion of external signals (irrespective of their form) leads to performance improvement over vanilla BART. This corroborates our discussion in previous studies that the impact of a contextual signal is not dependent on the signal but also on the underlying dataset itself. For both `SBIC` and `LatentHatred` *in-dataset* attributes (`Tox-BART`$_{\mathrm{C2}}$) perform at par with MIXGEN. The comparatively better performance of *in-dataset* against *in-domain* features reinstates the importance of human-in-the-loop to mitigate hatefulness. For `SBIC`, `Tox-BART`$_{\mathrm{C2}}$ displays comparable performance to both the MIXGEN setups – implicit and explicit knowledge with only a slight variation of (-2.27, -1.61, -5.13) and (+0.44, +1.83, -4.59) points in (BLEU, ROUGE-L, and BERTScore). In `LatentHatred`, `Tox-BART`$_{\mathrm{C2}}$ perform at par with MIXGEN with (BLEU, ROUGE-L, and BERTScore) scores of (1.44, -1.08, 0.8) and (0.49, -1.56, 0.77) for implicit and explicit knowledge baselines. With `LatentHatred` `Tox-BART`$_{\mathrm{C2}}$ beats the vanilla BART by (9.34, 17.05, 2.52) points in (BLEU, ROUGE-L, and BERTScore).

Table 5.7 also highlights that based on standard lexical metrics, zero-shot systems underperform. However, GPT-3.5 produces high semantic scores (> 90 BERTScores) for both datasets. We hypothesise this discrepancy in lexical metric arises as the train-test distribution for finetuned PLMs is closer than that for zero-shot GPT-3.5. Similar behaviour has been observed in other subjective tasks (Antoniak et al., 2024).

Ablations. We perform ablations on our *in-domain* setup (`Tox-BART`$_{\mathrm{C1}}$) using `SBIC`. In the first set of experiments, we alter `Tox-BART`$_{\mathrm{C1}}$ under various settings. In the first setting (**Exp. 1**), keeping all hyperparameters the same, we replace the toxicity tokens with pre-defined plain text, which is not a special token as provided in Table 5.8. From Table 5.9, we observe that a pre-defined text token as a feature significantly improves the results **(bold)**, bringing the performance closer to `Tox-BART`$_{\mathrm{C2}}$ and MIXGEN-*Exp*. In the second (**Exp. 2**) and third (**Exp. 3**) settings, we vary the threshold $\lambda = \{0.3, 0.6\}$. Though the difference is small, the default $\lambda = 0.5$ works best.

5.3.6 Error analysis

We discuss two classes of error observed with `Tox-BART`, provided in Table 5.10.

- **Modelling errors.** While training `Tox-BART`, we observe that the `SBIC` dataset has some empty rows (aka no gold explanations). For example, case #1 is hard to annotate without knowing if the question is out of curiosity, sarcasm, or disdain. Despite this, `Tox-BART` and even other baselines generate implied stereotypes,

Table 5.7: Results for generating explanations for implicit stereotypes for `SBIC` and `LatentHatred`. Bold (underlined) values represent the best-performing (second-best) setup for the given dataset for – B: max-BLEU; R: ROUGE-L F1; BS: BERTScore F1. For MIXGEN's implicit (explicit) signal infusion, we keep $k_i = 15$ ($k = 20$) as adopted from (Sridhar and Yang, 2022).

Method	**SBIC**			**LatentHatred**		
	B	**R**	**BS**	**B**	**R**	**BS**
GPT-2	62.72	62.72	59.04	30.94	21.99	82.71
BART	**72.17**	**70.83**	78.05	38.38	17.65	90.37
MIXGEN - *Imp*	72.12	69.84	80.91	46.28	35.78	92.09
MIXGEN - *Exp*	68.41	66.40	80.37	47.23	**36.26**	92.12
MIXGEN - *Exp* + *Imp*	70.27	67.69	80.23	47.00	33.09	90.8
Tox-BART$_{C1}$	64.89	63.83	64.52	41.94	26.28	89.47
Tox-BART$_{C2}$	69.85	68.23	75.78	**47.72**	34.70	**92.89**
GPT-3.5 (Zero-shot)	37.45	15.36	**90.10**	33.57	10.40	90.06

Table 5.8: Token to plain text mapping for ablation Exp 1.

Token	**Text**
$< TOXIC >$	toxic
$< NOT\_TOXIC >$	not toxic
$< SEVERE\_TOXIC >$	severely toxic
$< NOT\_SEVERE\_TOXIC >$	not severely toxic
$< OBSCENE >$	obscene
$< NOT\_OBSCENE >$	not obscene
$< IDENTITY\_ATTACK >$	identity attack
$< NOT\_IDENTITY\_ATTACK >$	no identity attack
$< INSULT >$	insulting
$< NOT\_INSULT >$	not insulting
$< THREAT >$	threatful
$< NOT\_THREAT >$	not threatful

leading to "hallucinated" explanations. Meanwhile, there are cases where the model failed to generate a contextual explanation, as highlighted in case #2. In #2, `Tox-BART` misses the climate reference. Lastly, we observe that in some instances, the PLM misidentifies the target and the subsequent explanation. For example, the focus on "nine-year-old" in case #3.

- **Annotation errors.** We notice that both `SBIC` and `LatentHatred` have mislabelling/incomplete explanations. For example, in case #3, some annotators provide incomplete sentences like "are losers" or phrases that can be triggering for the target group, like "everyone else is dead" with regard to school shootings. We also note that annotations can be highly subjective. For case #4, multiple stereotypes are true. In #4, the predicted stereotype, though valid, is not covered in the ground truth.

5.3.7 Can the quality of in-domain attributes impact `Tox-BART`?

Finally, we conclude our automated evaluations with an assessment of how the varying quality of toxicity attributes impacts performance.

Perturbing toxicity probabilities. Akin to perturbations in KG quality, we also

Table 5.9: Ablations on `Tox-BART`$_{\text{C1}}$ with `SBIC`. The first set adjusts the hyperparameters of C1 (Experiments 1-3). The second set perturbs toxicity probabilities (Experiments 4 a-c). The final set flips the attribute label (Experiments 5a-d). We report – B: max-BLEU; R: ROUGE-L F1; BS: BERTScore F1.

Method	B	R	BS
`Tox-BART`$_{\text{C1}}$	64.89	63.83	64.52
`Tox-BART`$_{\text{C1}}^{\text{Exp. 1}}$	**68.92**	**67.16**	**72.32**
`Tox-BART`$_{\text{C1}}^{\text{Exp. 2}}$	63.74	63.47	61.21
`Tox-BART`$_{\text{C1}}^{\text{Exp. 3}}$	62.8	62.76	59.34
`Tox-BART`$_{\text{C1}}^{\text{Exp. 4a}}$	63.16	62.96	60.11
`Tox-BART`$_{\text{C1}}^{\text{Exp. 4b}}$	62.95	62.82	59.72
`Tox-BART`$_{\text{C1}}^{\text{Exp. 4c}}$	64.73	64.25	62.79
`Tox-BART`$_{\text{C1}}^{\text{Exp. 5a}}$	65.16	64.24	64.87
`Tox-BART`$_{\text{C1}}^{\text{Exp. 5b}}$	**74.64**	**72.45**	**83.89**
`Tox-BART`$_{\text{C1}}^{\text{Exp. 5c}}$	63.55	63.36	60.63
`Tox-BART`$_{\text{C1}}^{\text{Exp. 5d}}$	63.9	63.45	61.85

Table 5.10: Some samples taken (verbatim) from `SBIC` and the predicted explanations by `Tox-BART`$_{\text{C1}}$ describing the error types. We also highlight whether the predicted model captures the target correctly for the gold target group. Empty strings (‘’) correspond to either no gold label or no generation by the model.

#	Utterance	Implied Stereotype	Target Group	Stereotype Prediction	Error Type
1	“Oh, you mean you’re Chinese, right?"	‘’	‘’	Asian folks are not smart	Hallucinating Stereotype
2	How’s America like the Dark Knight Trilogy? No one gives a fuck about Harvey.	trivializes harm to victims	climate disaster victims	‘’	Incomplete Stereotype
3	what do you call a nine-year-old with no friends? A sandy hook survivor	are losers.; everyone else is dead; school shooters are always socially isolated.	mass shooting victims	children are targets of pedos	Wrong Target Group + Incorrect/Incomplete Annotation
4	my friend thought she was lesbian just for having short hair. <repeat> “you can’t assume that," i said, “she’s not obese."	lesbian women are fat; lesbian women are unattractive	lesbian women	lesbians have short hair	Subjective annotations

experiment by perturbing the input probability scores (**Exp. 4**). We replace results from `ToxicBERT` by setting probabilities to a) all zeros, b) all ones, and c) random (between 0-1). From Table 5.9, we observe *the three adversarial configurations from **Exp. 4** register an expected deterioration in performance.* Albeit slight, the reduction in performance due to random perturbations in toxicity attributes reveals their domain-specificity in contrast to our observations with KG-tuples. These observations strengthen our initial decision to opt for toxicity attribute infusion instead of KG-tuples.

Impact of flipping attributes. We also measure the sensitivity of the model w.r.t flipped attributes (replacing A_i by $\neg A_i$). The results for the same are illustrated in **Exp. 5** of the Table 5.9. We perform these experiments for the top 4 attributes – toxic (**Exp. 5a**), severely toxic (**Exp. 5b**), obscene (**Exp. 5c**) and threat (**Exp. 5d**). We make an intriguing observation where flipping the severe toxic labels causes the model’s performance to overshoot well beyond the baselines. Since explaining implicit stereotypes aims to bring out the explicitness of a statement, we observe that highlighting an incoming post as extremely toxic nudges the model to produce more explicit explanations. *This uncanny observation calls into question the need for interoperability studies on how the augmentation of external signals nudge generations. We hypothesise that domain-specific* *generative PLMs are susceptible to extreme attributes from the same domain.*

5.3.8 Human evaluation as proxy content moderators

As the aim of the proposed setup is to enrich the content moderation pipeline and supplement moderator knowledge, we perform human evaluation employing humans as a proxy for content moderators. The human assessment further allows us to examine the semantic richness of `Tox-BART` and GPT-3.5 (the latter producing high semantic scores). Human evaluators are provided anonymised outputs from both systems against a given input sample, the gold explanation, and a gold target label. 20 evaluators access 17 random samples from `SBIC` on 5 metrics – *fluency, coherency, specificity, similarity with gold explanation, and target group*. *Fluency* and *coherency* measure the broader grammatical correctness. *Specificity*, *similarity with gold explanation*, and *target group* capture the task-specific correctness.

Evaluator recruitment. We recruit 20 human evaluators aged 18+ who have experience in using social media and work in CSS and NLP. The evaluation is voluntary, with no monetary compensation. It should be noted that while the initial shortlisting of samples to annotate is random, the final samples for evaluation are selected by the authors after vetting the initial text and its ground explanation (without looking at any model output) to minimise risk and harmful exposure for human subjects. We attempt to be as fair and diverse in our selection of samples as possible. Before the evaluation, we reached out to the people interested in participating. We give a detailed overview of the task (via email), providing them with material to sensitise them towards the task at hand. Further, the reviewers are known to have participated in some hate speech-related evaluations prior and have an idea about the content they would be engaging with. Only those willing to participate consensually are invited to participate in the review. Apart from the warning posted in the Google form, the evaluators are encouraged to contact the authors anytime during their evaluation to share feedback or discuss the content.

Evaluation guideline. The evaluators are provided with the following guidelines, and they can access the results independently for both systems.

Kindly go through the points below to gain context about the task before filling out the Google form. Filling out the form should not take more than 20-25 minutes. Thank you for your time!

Note. This form contains content that some might find offensive and upsetting. Reader discretion is advised. We begin by defining the basic terminology as follows:

- Stereotype. According to Wikipedia, a stereotype is referred to as "a generalised belief about a particular category of people."
- Stereotypical utterance. A stereotypical remark is an utterance that indirectly/implicitly hints at a stereotype.
- Implied stereotype. A short explanation in free text form of the stereotypical remark expressing the harmful and often offensive intent behind the remark towards the target group/category of people.

For each utterance, there are two machine generations for the implied stereotype expressing the intent behind the utterance. Each utterance will be referred to by the code

Ux, where x is the index number from 1-10, and the first generation by Sxa and the second generation by Sxb. For example, U3 refers to Utterance #3, S3a refers to the first stereotypical implication generation, and S3b refers to the second generation. For each generation, there are five metrics you will have to evaluate. We follow the 5-point Likert scale, with five being the highest. One metric is on a binary scale. You are required to compare each generation with the corresponding utterance and answer the questions that follow accordingly.

- Fluency. It measures how fluent the generation is in English, irrespective of its context regarding the task and its corresponding utterance. We only consider the syntactic properties of language here. Example: "My name is John" is a fluent sentence.
- Coherency. It measures how coherent the generation is. This is with respect to the utterance and the task. We aim to look at only the syntactic features via this metric. Example: Given an utterance that makes a stereotype against black folks indulging in criminal activities, the generation "this is a racial stereotype" is coherent with the utterance because it grabs the correct context regarding the utterance. Whereas a generation like "mentally disabled folks are dumb" is not because the original utterance is not talking about mentally disabled folks.
- Specificity. It measures how specific the generation is when considering the context of the utterance. This metric also determines how much contextually specific information is present in the generation. We aim to look at the semantic correctness via this metric. Example: For the same utterance as for the previous metric, the generation "this is a racial stereotype against black folks indulging in criminal activities" is much more specific than "this is a racial stereotype." Both generations might be equally coherent, but that does not imply how specific they are.
- Similarity with the gold explanation. It determines how similar the generations are with respect to any of the given gold annotations. You can combine your observations from metrics 2 and 3 here. Example: Given the gold label "racial stereotype against black folks indulging in criminal activities." The generation "this is a racial stereotype against black folks" is much more similar to the gold label than "this is a racial stereotype."
- Target group. It determines how correctly the generations identify the target group. You will be provided with the gold label and asked to mark whether the stereotype targets the same group. Option 0 [Target Not Correct] will be the valid option if the generation does not seem to target any group.

Observations. A manual analysis (Table 5.11) of the GPT-3.5-based generation reveals its tendency to produce non-specific/broad-stroke explanations. It may stem from GPT-3.5 being trained/filtered to discourage harmful discourse. Our investigation aligns with practitioners' observations that GPT-3.5-based LLMs are rigorously guardrailed, hampering their ability to perform well in tasks such as ours, necessitating the generation to be explicit and specific about stereotypes. For example, looking at the first instance in Table 5.11, we see that the terms "racist stereotype" and "black people" are semantically close to the gold generations, even though it is not specific. Since BertScore (Zhang* *et al.*, 2020) employs pair-wise semantic embedding matching, using generic terms that are semantically closer to the target group helps GPT-3.5 maintain the high BertScore. Yet, it leads to higher variability on *Specificity* (Table 5.12 (a)) for GPT-3.5.

Table 5.11: Some samples taken (verbatim) from SBIC and their predicted explanations by Tox-BART$_{C1}$ and GPT-3.5.

Utterance	Tox-BART$_{C1}$	GPT-3.5 (zero-shot)	Gold Label Stereotype(s)
You used to be able to walk into a shop with £2.00 and leave with all you could need. Now, all the blacks are beating us to it.	black people are thieves	Racist stereotypes about black people.	black people steal, blacks are getting there before whites, there are too many black people, blacks are beating whites to the punch, black people are greedy, black people are always taking things
What's the difference between science and religion? Science will fly you to mars. Religion will fly you into a skyscraper	muslims are terrorists.	Science = Mars, Religion = Skyscraper.	islamic folks religion is only about hurting people, religion is bad, islamic folks religion is a joke, religion makes you stupid
Why are all women left handed? Because they don't have rights.	women don't have rights	Women are not capable.	women are slaves, women are inferior, don't have all rights men have, women are stupid, are missing rights that they should have, women have no rights
Dicks are basically ar-15s. Only disabled kids whip them out in class,	physically disabled folks are stupid	Ableist stereotypes.	don't have control of themselves, show their private areas in public, mentally disabled folks are violent, mentally disabled folks are stupid

Table 5.12: Comparison between Tox-BART$_{C1}$ and GPT-3.5 on 17 random samples of SBIC via – (a) Mean (std) score for each metric from 20 evaluators. Fluency (Flu.), coherence (Coh.), specificity (Spe.), and gold similarity (Sim.) are on a 5-point scale, and the target detection (Tar.) is boolean. (b) Toxicity scores from the Unitary toxicity API. The higher the toxicity, the closer to the intended explicit connotation of the explanations.

(a)

Method	Flu.	Coh.	Spe.	Sim.	Tar.
Tox-BART$_{C1}$	4.52 (±0.76)	3.95 (±0.99)	3.67 (±0.92)	3.47 (±1.00)	0.78 (±0.29)
GPT-3.5	4.17 (±0.9)	3.74 (±0.92)	3.27 (±1.07)	2.78 (±1.14)	0.49 (±0.4)

(b)

Method	Toxicity↑
Tox-BART$_{C1}$	0.89 (±0.21)
GPT-3.5	0.33 (±0.32)

To clarify, we did not explicitly prompt any system to predict the target. Instead, human evaluators determine if the model under evaluation can mark the correct target group within the explanation it generates. Here, human evaluators find that 49% of the time, GPT-3.5 focuses on either the wrong target group or talking about the wrong stereotype for the given target group. In comparison, Tox-BART is able to refer to the correct target group 78% of the time. We want to point out that the target group specified in both datasets is annotated by humans in the respective datasets in a free text form, leading to some raw 800 different target names. A categorical detection and assessment is not feasible. Hence, our reliance on human evaluation. We further corroborate the generality of the explanations from GPT-3.5 by computing toxicity scores from Unitary toxicity API (Hanu and Unitary team, 2020).

On average, Tox-BART$_{C1}$'s generations are much more toxic compared to GPT-3.5 (0.89 vs. 0.33), as observed in Table 5.12 (b). As we aim to unmask the underlying stereotype, the generated output is expected to be explicit.

5.3.9 Limitations and future work

- **Limited datasets in non-English.** Stereotyping and implicit hate datasets that capture contextual and cultural nuances beyond English (West) are largely missing.
- **Social biases.** Any toxicity analysis systems (whether classification or generation) suffer from social biases they learn from the extensive pretraining corpus and the subjectivity of the annotated downstream tasks (Garg *et al.*, 2023). It can be destructive in the long run (Gehman *et al.*, 2020).

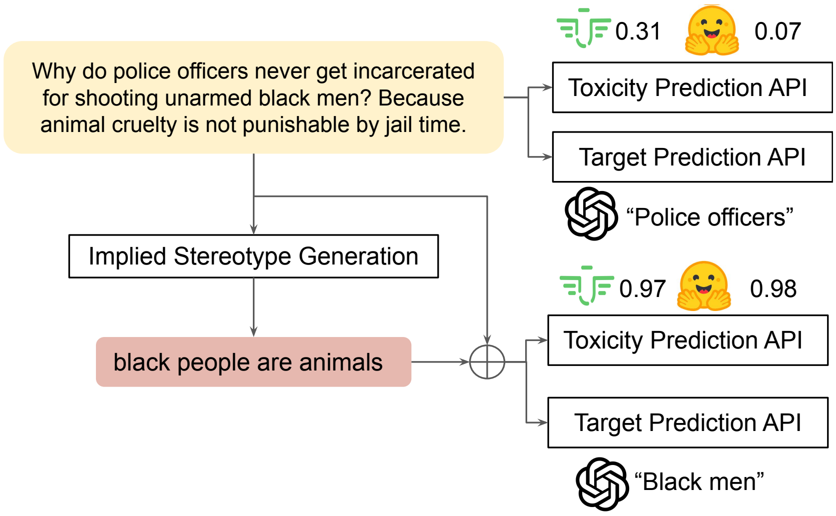


Figure 5.5: A sample text (verbatim from SBIC) witnessing an improvement in toxicity and target detection when the incoming post is infused with implied context. We infer toxicity scores from the Unitary toxicity API and Toxigen-RoBERTa. For target detection, we prompt the ChatGPT user interface.

- **Incomplete covering and predictions.** Human annotations not encompassing all viewpoints of the target group provide incomplete perspectives to the model. Further, given the implicit nature of the task, the proposed system may miss out on correctly identifying instances of sarcasm and irony. Incorrect identification of the target group or propagation of hallucinated stereotypes is equally problematic.

5.4 Chapter conclusion

Implicit hate speech is challenging to unmask. We observe this even with sophisticated systems like ChatGPT (GPT-3.5). The efficacy of content moderation improves when the implicit hate is accompanied by its underlying explicit explanation, as outlined in Figure 5.5.

However, our counter-intuitive observations and an examination of general-purpose vs domain-specific KGs highlight the issue of signal/noise in the retrieved tuples for explaining implicit hate. The absence of explicit hate or indirect mention of the target means that extracted entities may not relate to hateful connotations. Having established the (ir)relevance of commonsense knowledge-based systems, we examine the efficacy of *in-domain* and *in-dataset* toxicity features for the task of explaining implicit hate. The more straightforward and explicit (and therefore seemingly toxic) the explanations, the better equipped the content moderators (humans or computers) will be to judge the incoming implicit hate. It is important to reiterate that an increase in explicitness comes at the cost of specificity. *Tox-BART can achieve optimal performance in balancing the explicitness while retaining the specificity of the target group and the underlying stereotype.*

It is essential to point out the dependency of the proposed model on the external toxicity signal (either manually annotated or obtained from an already finetuned end-point). Our error analysis highlights that subjective tasks mitigating toxicity cannot be fully automated. Here, the way forward is a human intervention to compile the final version of machine-generated labels and context. Future works must also focus on developing datasets and systems to enable social reasoning (Zhou *et al.*, 2023) and reduce the inference cost of incorporating external signals by continued pretraining.

CHAPTER 6

Proactive Hate Mitigation

"Everyone has the right to refute any opinion. But no one has the right to prevent its expression."
— Periyar E.V. Ramasamy

6.1 Chapter introduction

Content moderation systems, whether supported by classification or generative tasks, result in reactive (after a post has gone public) flagging/removing of content or banning users, or both. However, despite intentionally engaging in hateful content, a section of online users are adaptive and can be nudged to change their opinions via empathy (Hangartner *et al.*, 2021) and corrective behaviour.

An alternative solution can be to proactively counter hate speech before it goes public (Chaudhary *et al.*, 2021). Systematic and proactive sensitisation of online users can help them voice their opinions without directly propagating harm (Dementieva *et al.*, 2025). In this case, promoting users with alternative scenarios nudges a change in attitude (Kaufman and Libby, 2012; Bhuiyan *et al.*, 2018). Twitter[1] and Instagram[2], among other platforms, have already rolled out the experimental prompts for 'offensive posts with an option to revise before going public'. Based on experiments with such prompts, researchers at Twitter published a pilot study that prompted participants to eventually post fewer offensive tweets than non-prompted users in the control group (Katsaros *et al.*, 2022). Such nudging should be more welcoming to the users than them getting banned from the platform without having an opportunity to improve.

In this chapter, we continue to work towards user-centric tooling with the introduction of a novel task, *hate speech normalisation*. It aims to weaken the intensity of hatred exhibited by an online post. The intention of hate speech normalisation is not to support hate but instead to provide the users with a stepping stone towards non-hate, while giving online platforms more time to monitor any improvement in the user's behaviour. To this end, we manually curate a parallel corpus `HateNorm` (**H**ate **N**ormalisation) – hate texts and their normalised counterparts (a *normalised text* is less hateful and more benign). We introduce **N**eural h**A**te spee**Ch** norma**L**iser (`NACL`), a simple yet efficient hate speech normalisation model. `NACL` yields a score of 82.27 BLEU and 80.05 perplexity for the normalised text generation. An interactive prototype of `NACL` is put together for the user study. Continuing to explore the "toxicity attributed" to contextual signals, in this chapter, we depend on 'hate intensity' to nudge the generations from the PLM.

[1] https://www.socialmediatoday.com/news/twitter-is-testing-prompts-which-would-recommend-users-hide-potentially-off/586872/

[2] https://www.socialmediatoday.com/news/instagram-adds-new-anti-bullying-measures-including-comment-warnings-and-u/558307/

6.2 Reducing the intensity of hate via normalisation

6.2.1 Motivation

Research gap. There will always be users who intentionally and constantly spread hateful content. A cross-platform study of hateful users revealed how the banning of users could backfire for the platform and the Internet community at large (Johnson *et al.*, 2019). A blanket ban on such activities is infeasible due to several geopolitical and cultural reasons. Meanwhile, techniques for paraphrasing offensive text suffer from major shortcomings (Nogueira dos Santos *et al.*, 2018; Tran *et al.*, 2020). They aim to convert offensive text to non-offensive/neutral. Prompting users to change their texts to be completely non-hateful can be a significant behavioural shift and may backfire. Extending upon the idea of proactive nudging (Katsaros *et al.*, 2022), we hypothesise that a counterpart of hateful content suggested by weakening the overall hatred while retaining the underlying semantics can be the stepping stone instead.

Research questions. Through a proactive moderation setup, we aim to examine:

RQ1: What is the impact of normalisation in driving the post's engagement?
RQ2: How can PLMs be contextually nudged to generate low-intensity hate?
RQ3: What human evaluations can be conducted to corroborate nudging users in real-time towards less hatefulness?

Contribution summary. Our aim is not to render a hateful message into a non-hateful one; instead, it is the reduction of hatred. To this end, we manually curate a parallel corpus of hateful samples and their normalised counterparts, the first dataset of its kind. Our preliminary analysis of the `HateNorm` dataset suggests that, in general, only a few phrases within a sentence convey significant hatred. Therefore, we first identify the hateful spans for each hateful sample and then normalise them to reduce overall hatred. We propose `NACL`, which operates in three stages. It first measures the intensity of hatred exhibited by an incoming sample by employing an attention-driven Bidirectional-LSTM (BiLSTM) regression model. Following this, a BiLSTM-Conditional random field (CRF) module operates on the sample to extract the hateful spans from the source sample. In the last stage, we incorporate BART to normalise the incoming samples. The penalty from the intensity discriminator (BiLSTM-regressor) enforces the generator (BART) to normalise the hateful samples. We obtained a performance of 0.136 RMSE, 0.622 F1-score, and 82.27 BLEU in intensity prediction, span identification, and text normalisation, respectively. We further show the generalisability of `NACL` across three social media platforms – Reddit, Facebook, and Gab. We develop an end-to-end system that can be efficiently deployed in a real-world environment. Overall, we observe that `NACL` reports coherent and better outputs as evaluated by humans (Masud *et al.*, 2022).

6.2.2 Dataset curation

We source and compile a list of hateful instances by restricting our samples to the hateful labels of the multiple English datasets (Basile *et al.*, 2019; Davidson *et al.*, 2019; de Gibert *et al.*, 2018; Chung *et al.*, 2019; Mathew *et al.*, 2021; Jha and Mamidi, 2017). In total, we collect 4423 hateful samples and annotate them with hate intensity scores and hateful spans. At the end of the process, we observe 1396 samples to have no explicit spans, have the whole post as a span or exhibit hate intensity less than the threshold τ. Thus, our final gold dataset consists of 3027 parallel hateful and normalised samples, along with the intensity scores and hate spans in the original ones (Table 6.2). Additionally, we annotate each normalised sample with the updated hate intensity score.

Table 6.1: Data samples with hate intensity and spans. Intensity scales from $[1-10]$ (a higher score denotes a higher degree of hatefulness in the sample). The phrases in blue denote the hate spans in the original sentence. We also provide the normalised versions of the samples. Section 6.2.2 describes the annotation scheme and hate intensity score in detail (**Int:** Intensity).

Type	Sample	Int
Org	*This {immigrant should be hung or shot ! Period! An***}$_{Span}$. @user*	8
Norm	This immigrant should be punished. @user	4
Org	*@user This {trashy wh*** gold digger}$_{Span}$ at again trying to be something that makes her think shes good ughhh not happening {h** bag pos}$_{Span}$*	9
Norm	@user This money loving women at again trying be something that makes her think shes good ughhh not happening	2
Org	*{#BuildThatWall #BuildTheDamnWall #BuildThatWall BUILD THAT EFFEN WALL}$_{Span}$ I'm sorry my Lord #Jesus but people are just deaf down here*	8
Norm	Ask them to leave the nation. I'm sorry, my Lord Jesus, but people are just deaf down here	4
Org	*Women ... Can't live with them ... {Can't shoot them}$_{Span}$*	6
Norm	Women ... Can't live with them ... can't speak to them	3

Table 6.2: Dataset statistics of `HateNorm`.

Statistics	**Value**
Total samples	4423
Sample length	23 (avg), 112 (max)
No. of samples with intensity scores	4423
Hate intensity range	1 (min) – 10 (max)
No. of samples with spans	3027
No. of spans	5732
Avg length of spans (tokens)	3
No. of normalised samples	3027
Normalised sample length	21 (avg), 99 (max)

6.2.3 Dataset annotation

For annotations and human evaluation, the input texts are not masked and presented as is. Table 6.2 lists the overall statistics of the data curated and annotated. Table 6.1 shows a few examples of the original hate samples, along with their normalised forms and intensity scores for both sets. A bar graph representing the original and normalised hate intensity distributions is shown in Figure 6.1. *The intensity distribution mass in original samples shifts towards the weaker intensity in normalised samples.*

Annotator details. We recruit two annotators for the annotation process. Both annotators annotated all the available samples. In case of disagreement (11% of cases), we employ a third annotator to break the tie. Overall, the mean squared error IAA for hate intensity is 0.20. In case of a disagreement on the hate span, if the non-overlap contains words like abuse or racial slurs, we add them to the span. In rare cases, if both annotators and third annotators disagree over some time, the sample is dropped from the final dataset.

Annotation guidelines. For this setup, we follow the definitions proposed by

Waseem and Hovy (2016) for hate speech and mark the hate span if it consists of any of the following explicit mentions:

- A sexist or racist slur term or an abusive term directly attacking a minority group/individual.
- A phrase that advocates violent action or hate crime against a group/individual.
- Negatively stereotyping a group/individual with unfounded claims or false criminal accusations.
- Hashtag(s) supporting one or more of the points mentioned earlier.

Additionally, the hate intensity of a sample is marked on a scale of $1 - 10$, 10 being the highest based on:

- Score$[8 - 10]$: The sample promotes hate crime and calls for violence against the individual/group.
- Score$[6 - 7]$: The sample is mainly composed of sexist/racist terms or portrays a sense of gender/racial superiority on the part of the person sharing the sample.
- Score$[4 - 5]$: Mainly consists of offensive hashtags, or most hateful phrases are in the form of offensive hashtags.
- Score$[1 - 3]$: The sample uses dark humour or implicit hateful terms.

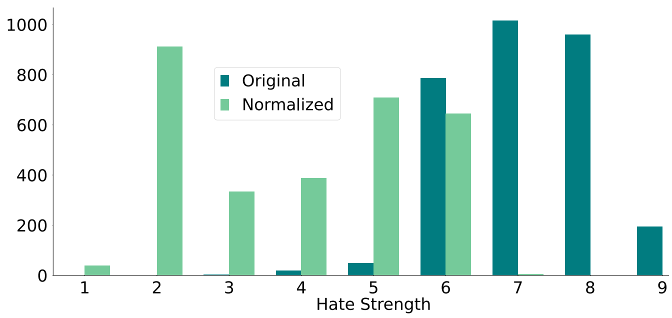


Figure 6.1: Hate intensity distribution for all the original and normalised samples. The distribution shifts towards the weaker intensities under normalisation.

6.2.4 Can normalised posts reduce engagement?

A significant criticism of prompting and suggestive rectification is the debate about the reduced expressive powers of the users. However, the study on Twitter revealed that encouraging users to minimise the publishing of offensive text has no significant impact on their ability to participate in non-offensive conversations (Katsaros *et al.*, 2022). This result is a big motivation for researchers exploring proactive methods of countering hateful/hurtful content, as we do in this chapter.

Hypothesis. Though one can argue that normalised text is still hateful and prone to spreading harm, we hypothesise that in its normalised form with a reduction in intensity, the content should see a decline in user engagement (i.e., reduction in virality).

Setup. We take inspiration from the virality prediction models on social media to test this hypothesis. In our setting, the virality of a post is expressed in terms of the total comments it receives. Since ours is a text-only dataset curated from different sources,

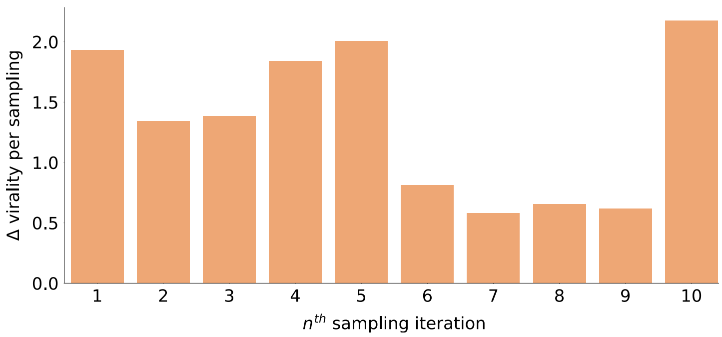


Figure 6.2: Difference in the predicted number of comments per set per iteration. During each iteration, we obtain the median difference (Δ) in the predicted number of comments for hateful and corresponding normalised sampled pairs.

we lack the availability of any network, temporal and other signals. Engineering various textual features (listed below) inspired by the work of Dutta *et al.* (2020) and Cheng *et al.* (2014), we train a user engagement prediction model (trained on the Reddit dataset). We use it to predict the comment count for the hateful posts and their normalised counterpart present in `HateNorm`. We use hateful and normalised samples as our control and treatment sets, respectively. We randomly sample 300 pairs of test cases for 10 iterations and record differences in median comment count per sampling iteration. The following text-based features are employed for predicting comment engagement/virality:

- Complexity. It measures the uniqueness of terms introduced as the logarithm of term frequency in the test samples.
- Readability. It is computed via LIX and RIX readability scores[3].
- Informativeness. It is obtained by the summation of the TF-IDF vectors of the words in the sample.
- Polarity. It is computed as the overall SentiNet[4] polarity score of the sample.

Observations. From Figure 6.2, we find the phenomenon (Δ in the median virality of hate vs normalised) to be statistically significant with a p-value of 0.0027 and effect-size of 2.324 on Welch's t-test. This evidence is in line with existing studies (Katsaros *et al.*, 2022), which find that less offensive content is expected to catch fewer eyeballs.

Through randomised experiments with engagement prediction, we conclude that even if present online, the normalised form of a post is less likely to gain engagement. We envision that proactive nudging can allow for varying sentiments (including hateful ones) to coexist with neutral content, balancing both freedom of speech and censorship.

6.2.5 Design of NACL

We engineer a real-time deployable system that can normalise hate speech on the fly. Figure 6.3 illustrates a high-level overview of the `NACL` framework. We assume that `NACL` will be called to action once a text is detected as hateful by a preexisting hate

[3] https://readable.com/blog/the-lix-and-rix-readability-formulas/
[4] https://pypi.org/project/SentiNet/

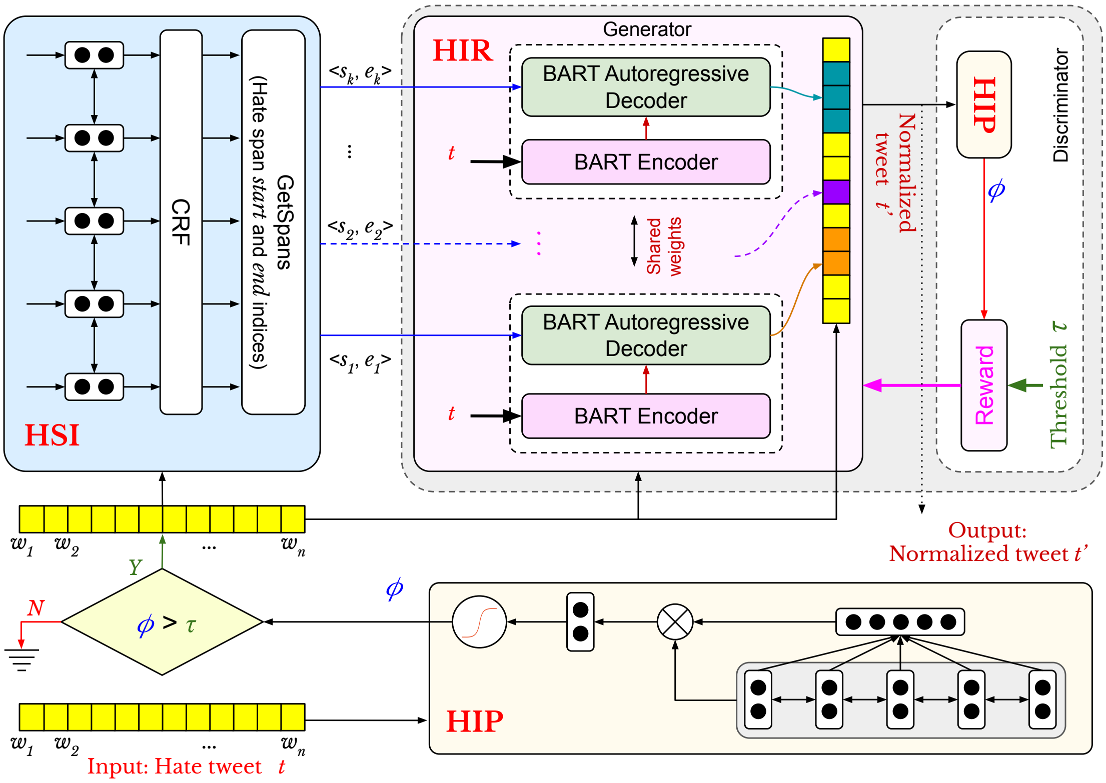


Figure 6.3: A schematic view of NACL. The first stage of the model (i.e., HIP) validates the eligibility of a sample for normalisation. Subsequently, HSI identifies the start and the end indices to mark hate spans in eligible samples. Finally, for each identified hate span, a normalised paraphrase is generated by HIR and validated by the discriminator to ensure intensity reduction.

detection model. In our case, we mimic it by only using hateful samples for training NACL. Any future reference to the input sample would mean that it is already labelled as hate. The pseudo-code is summarised in Appendix H.

Problem statement. For a given hate sample t, our objective is to obtain its normalised form t' such that the intensity of hatred ϕ_t is reduced, i.e., $\phi_{t'} < \phi_t$. To achieve this, we divide the overall task into three stages. We compute the hate intensity (ϕ_t) of an input sample t and forward it for normalisation only if it satisfies the condition, $\phi_t > \tau$ (i.e., only strongly hateful samples are subject to normalisation).

Hate intensity prediction (HIP)

Objective. The task of NACL-HIP is to determine the extent (degree or intensity) of hatred in a message (inspired by the toxicity API[5]). Given a sample t, it measures the intensity of hatred on a scale of $[1, 10]$, with 10 being the highest, i.e., $\phi_t = HIP(t)$.

Formulation. For HIP, we employ a Bi-LSTM regressor. The intermediate representation is passed through a self-attention layer connected to a fully connected linear activation. The embeddings are obtained from a BERT model.

[5]https://www.perspectiveapi.com

Hate span identification (HSI)

Objective. Hateful spans in a sentence are the portions of a sentence responsible for conveying hate (Pavlopoulos *et al.*, 2021). The intuition behind this task is that we can achieve sentence-level normalisation if we attempt to normalise these spans. Given a hateful sample, tokenised as $t = \langle w_1, w_2, \ldots, w_n \rangle$, we look for a consecutive sequence of hateful tokens, $\langle w_i, \ldots, w_{i+l} \rangle$. *A sample can have multiple, non-overlapping hate spans, and $HSI(t)$ aims to find all such spans.*

Formulation. Our goal is to train a model to detect relevant hate spans represented by tags (B, I, O) where 'B' represents the beginning of a hate span, 'I' forms the continuation of a hate span, and 'O' represents the non-hateful spans. For our model, we again use Bi-LSTM to capture the contextual representation of sequences. The hidden representations are then passed through a time-distributed dense layer to flatten the embedding structure. We further use a Conditional Random Field (CRF) layer to fit our representations and produce the required tags for our sequences. The CRF layer models the conditional probability of each sequence $p(s_1, ..., s_m | w_1, ..., w_m)$ by defining a trainable feature map $\phi(w_1, ..., w_m) \in R^d$ that maps an entire sequence to a h dimensional feature vector. Here $s_i \in \{B, I, O\}$. We can model the probability using the feature vector $h \in R^d$ as Equation 6.1 describes:

$$p(s|w; h) = \sum_s exp(\phi(w, s)), \tag{6.1}$$

After getting the feature vector h^*, we can find the most likely tag for a sentence s^* by: $s^* = argmax\ p(s|w; h*)$

Hate intensity reduction (HIR)

Objective. The objective of `NACL`-HIR is to generate a new sample that preserves the original semantics but with weaker hate intensity. Intuitively, we first need to identify the threshold for strong vs weak hate. For each strong hate sample with $\phi_t > \tau$, we aim to generate semantically similar sample $t' \approx t$, with the constraint $\phi_{t'} \leq \tau$.

Formulation. The hate intensity reduction model is a Generative Adversarial Network (GAN) based architecture, which employs both `NACL`-HIP as its assistive modules. `NACL`-HIR accepts a hateful sample along with its span labels as identified by the HSI model. For `NACL`-HIR, we use a pretrained BART to generate the normalised spans based on the hate spans identified by the `NACL`-HSI module. The generated spans are amalgamated with the rest of the sample tokens and forwarded to the HIP-based discriminator model. The discriminator computes the hate intensity score $\phi_{t'}$ of the normalised text t'. Since our objective is to reduce the intensity of t', we assign a reward/penalty R to the generator based on the hate intensity score $\phi_{t'}$ of the normalised sentence and the accepted threshold value τ as via Equation 6.2:

$$R_{t'} = \tau - \phi_{t'} \tag{6.2}$$

The generator consumes the discriminant reward into its loss function as follows Equation 6.3:

$$\mathcal{L} = \ell + (1 - R_{t'}) \tag{6.3}$$

Here ℓ is the generator loss. Backpropagation aims to minimise the consolidated loss $\mathcal{L}$ in order to generate semantically coherent normalised samples with hate intensity less than or equal to τ. Note that the threshold value is a hyperparameter. If the computed hate intensity score is greater than the threshold, the negative reward penalises the generator to improve its prediction. It is interesting to note that normalised paraphrasing can be obtained only with HIP and HIR, as the non-hateful spans will not contribute to intensity in the latent space. We iteratively build our modules, starting independently with the HIP and HSI modules and then using only the best-performing ones for HIR.

6.2.6 Experimental setup

Train and test sets. We split `HateNorm` in the $70:15:15$ for the training (2119), validation (454), and testing (455). One may rightfully argue that such a small number of samples in the test set (S_A) may not be adequate to evaluate the modules. Therefore, we compile another set of $1,111$ hate samples from two additional sources (He *et al.*, 2022; Gao and Huang, 2017). These are marked as the secondary test set (S_B). Note that samples in S_B do not have gold-normalised counterparts.

Evaluation metrics. We compute Pearson correlation, cosine similarity and RMSE scores for HIP. To capture the sequential nature of BIO tags, we employ the seqeval macro-F1[6] for HSI. In the case of HIR, which is a generative task, we compute perplexity and BLEU scores. Note that given the n-gram overlap and change in hatefulness between original and normalised sentences, semantic metrics are not used.

Hyperparameters. We experiment with two threshold values ($\tau = \{3, 5\}$) for the hate normalisation. The models make use of Tensorflow 2.0 and Transformer 4.5.1 with Python 3 libraries, trained on Google Colab with Tesla P100-PCIE-16GB GPU. For the `NACL`-HIP model, we employ a single Bi-LSTM layer (*hdim*=512) followed by a layer of self-attention. For this model, the MSE loss and the Adam optimiser with linear activation are used. The input embedding is 'bert-base' with *dim*=768. The model is trained for 10 epochs with a batch size of 32. For `NACL`-HSI, we employ a 2 layer Bi-LSTM (*hdim*=512), using a batch size of 32, and the RELU activation function. As the final layer for the Span Model is a CRF, we take 'crf loss' and 'crf accuracy' as our loss and accuracy metric, respectively, and use the Adam optimiser. The model is trained for 5 epochs with a batch size of 32. For `NACL`-HIR, we employ a 'bart-base' with RELU activation and Adam optimiser.

Baselines. The setup lacks exact baselines due to the novelty of the proposed task. Therefore, we adopt six existing methods as baselines pertinent to our work.

- **Dictionary-based normalisation.** Inspired by the early work of Su *et al.* (2017), we define a dictionary-based model that learns the mapping between the hateful span and its corresponding normalised span from the training set. We employ TF-IDF-based cosine similarity to perform a lookup to select the closest normalised span for the hate span in the test sample.
- **Neutralising subjective bias.** This model aims to counteract the subjective bias in news content by converting a piece of opinionated news into a neutral one (Pryzant *et al.*, 2020). It utilises an LSTM-based model. We retrain the model for the hate

[6] https://github.com/chakki-works/seqeval

normalisation task, assuming that strongly hateful content is also biased. Note that this method does not guarantee coherence in replaced spans.

- **FGST**: Given a sentence with a polarity label (*positive* or *negative*), FGST (fine-grained text sentiment transfer) (Luo *et al.*, 2019) generates a new sentence having a pre-defined sentiment intensity score. In our setting, we aim to develop a normalised text having $\phi_t \leq \tau$.
- **Style transfer models.** For these unsupervised models, we retrain the model for style transfer between original and normalised sentences.
  - **Style transformer.** It is a transformer-based architecture which generates a new sentence (for a given source sentence) without making any assumption about the latent representation of the source sentence (Dai *et al.*, 2019).
  - **NPTCA.** It generates a new sentence for a chosen sentence by assuming that different corpora possess a shared latent content distribution (Shen *et al.*, 2017). It uses two constrained versions of auto-encoders (aligned and cross-aligned) to refine the alignment of these latent spaces.
  - **GST.** For a sentence, GST (Generative Style Transformer) (Li *et al.*, 2018) generates a new sentence by modifying a specific attribute, such as sentiment, while keeping other attributes intact. It deletes phrases associated with the attribute in the original text and then retrieves new phrases related to the target attribute and uses a sequence-to-sequence model to combine them.

6.2.7 Results

For automatic evaluation, we employ a two-way setup. First, we perform an intrinsic assessment to measure the model's performance using standard evaluation metrics. Second, we employ hate speech detection models to discriminate between the original and normalised hate samples for extrinsic evaluation.

Intrinsic evaluation

Here we report the performance on standard evaluation metrics for each of the three NACL components.

Table 6.3: Results for hate intensity prediction (HIP).

Model	Embedding	Evaluation Measure		
		Pearson ↑	**Cosine Sim ↑**	**RMSE ↓**
CNN	GLV	0.2827	0.2613	3.175
BiLSTM	GLV	0.1939	0.2044	3.411
BiLSTM+CNN	GLV	0.3600	0.3124	4.92
CNN	BERT	0.2449	0.212	2.9823
BiLSTM	BERT	0.2613	0.2457	4.9414
BiLSTM+CNN	BERT	0.3211	0.3375	3.691
BERT		0.5558	0.5927	1.7712
NACL-HIP	ELMO	0.4521	0.4141	0.9982
NACL-HIP	BERT (linear)	**0.766**	**0.973**	**0.136**
NACL-HIP	BERT (sigmoid)	0.704	0.968	0.148

HIP. As shown in Table 6.3, using BERT+BiLSTM with a linear activation gives the best results with a cosine similarity of 0.973, Pearson score of 0.766 and an RMSE of

0.136. Meanwhile, BERT+LSTM with a sigmoid activation (scaling intensity between 0-1 range) reports a cosine similarity of 0.968, Pearson score of 0.704 and RMSE of 0.148. Overall, the linear activation beats sigmoid; for the rest of our experimentations, we use BERT+LSTM (linear) as our HIP model.

HSI. BiLSTM+CRF yields the best macro-F1 of 0.622 with ELMo embeddings as reported in Table 6.4. Among others, the finetuned SpanBERT model stands second, with 0.583 macro-F1. For the rest of our experimentation, we use ELMo+BiLSTM+CRF as our HSI model.

Table 6.4: Results for hate span identification (HSI).

Model	**Embedding**	**Evaluation Measure**		
		Precision ↑	**Recall ↑**	**F1 Score ↑**
CRF	GLV	0.7013	0.3867	0.4985
CRF	BERT	0.6624	0.3335	0.4437
BERT		0.6053	0.3676	0.4574
SpanBERT		0.7081	0.5413	0.6135
NACL-HSI	GLV	0.491	0.458	0.470
	BERT	0.6471	0.4823	0.5526
	ELMO	0.619	**0.634**	**0.622**
	SpanBERT	**0.6913**	0.5041	0.5830

Table 6.5: Results for hate intensity reduction (HIR).

Supervised	**Model**	**Evaluation Measure**	
		BLEU ↑	**Perplexity ↓**
Yes	Dictionary Model	55.18	92
	Bias Neutralisation	39.5	90.38
No	FGST	39.4	123.38
	Style Transformer (ST)	15.6	200.85
	Style Transfer (NPTCA)	0.93	1138.4
	Style Transfer (GST)	0.84	199.58
Yes	NACL-HSR (τ=3)	58.9	86.11
	NACL-HSR (τ=5)	**82.3**	**80.05**
	Gold	100	64.66

HIR. Finally, we report the performance of NACL-HIR module along with other baselines in Table 6.5. NACL-HIR yields the best perplexity score (a lower value is better) of 80.05 for the generated normalised sentences. In comparison, the dictionary-based baseline obtains the perplexity of 92. For reference, we also compute perplexity (64.66) for the gold normalised sentences. Moreover, we observe a similar trend in the BLEU scores as well. The NACL-HIR model achieves the highest BLEU (82.27). The high BLEU score can be attributed to the fact that, due to the intensity-based reward, HIR mainly targets hate spans for normalisation, and a good portion of the original token (not containing hate) sequence gets preserved in the normalised sentence. We observe that methods like Bias Neutralisation and FGST perform moderately well in comparison to NACL-HIR. In contrast, we observe abysmal performance from the style transfer methods. This is mainly due to the lack of a large-scale corpus to train these models for our problem definition. During the human evaluation, we do not consider such baselines.

Table 6.5 shows that between $\tau = 3$ and 5, the latter is better suited for the NACL-HIR model. Therefore, for the rest of the evaluations, we work with $\tau = 5$ as the threshold.

> Interestingly, the use of hate intensity as a reward for NACL corroborates the use of toxicity-attributed signals employed in the last chapter (Yadav *et al.*, 2024).

Extrinsic evaluation

We hypothesise that a hate speech detection method will exhibit less confidence in classifying the normalised text as hate. It is the first step toward extending the evaluation for adversarial attacks, which we hope to study in the future. Of extrinsic assessment, we employ three widely-used hate speech detection methods by Waseem and Hovy (2016), Davidson *et al.* (2019), and Founta *et al.* (2019). We train the methods on their respective datasets (outlined in Table 6.6). For consistency, we map multiple granular hate labels into *hate* and *non-hate* labels.

Evaluation on Test Sets $\mathbf{S_A}$ and $\mathbf{S_B}$. For each original sample t in $S_A + S_B$, we extract $\gamma(t, m)$, the softmax probability of the *hate* class as the confidence score, where m is the underlying hate detection model. Evidently, $\gamma(\cdot) \in (0, 1]$. Subsequently, we compute the confidence score, $\gamma(t', m)$ for each generated normalised sample t'. Considering that hate speech normalisation aims to reduce the hate intensity of samples instead of converting to non-hate, we restrict ourselves to the set of samples in $M_m \subseteq S_A + S_B$ for which $\gamma(t) \geq 0.5$ and $\gamma(t') \geq 0.5$, i.e., both the original and normalised samples are predicted as hate. Finally, we compute the average difference in confidence score, Δ_c, for each pair (t, t') in M_m via Equation 6.4.

$$\Delta_c(t, t') = \frac{1}{|M_m|} \sum_{t \in M_m} \gamma(t) - \gamma(t') \tag{6.4}$$

Table 6.6: The hate vs non-hate tweet ratio for the hate classification. The data is collected using the respective methods.

Model	# of tweets	Hate:Non-Hate
Waseem (Waseem and Hovy, 2016)	12772	4750:8022
Davidson (Davidson *et al.*, 2019)	24783	20620:4183
Founta (Founta *et al.*, 2019)	59189	13551:45638

Table 6.7: Average change (Δ_c) in confidence of the predicted hate class of three hate detection methods. Δ_c is computed between the original and normalised pairs in the test set against the respective normalisation model (Equation 6.4).

	Normalisation model's Δ_c (↑)					
Hate detection method	**FGST**	**Bias**	**ST**	**GST**	**NPTCA**	**NACL-HIR**
Waseem and Hovy	0.00	0.03	0.03	−0.02	−0.04	0.03
Davidson	0.04	0.00	0.00	0.35	0.21	0.26
Founta	0.04	−0.01	0.07	0.23	0.04	0.03

Table 6.7 reports the average difference in confidence scores for NACL-HIR and other baselines. Since we can't generate the dictionary-driven normalised samples for the test set in S_B in the absence of hateful spans, we do not include it in our extrinsic evaluation. We observe the consistency of NACL-HIR over other normalisation models as

it obtains confidence scores > 0 across all three hate detection methods. In comparison, Style Transfer GST gives the highest performance in two cases but reports a negative score in the third. It should be noted that all existing hate speech datasets are highly skewed against hateful labels (Founta *et al.*, 2018). Thus, detection models are prone to underperform for the hate class. Having considered that, extrinsic evaluation aimed at pointing out the general pattern in the reduction of hatefulness post-normalisation. Such behaviour is expected to manifest irrespective of the hate detection model employed.

Table 6.8: Gold and predicted intensites for NACL-HIP.

#	Example	Gold	Pred
1.	b***h you're a grown a** women @user	10	10
2.	@user happy birthday qu**r	8	4
3.	@user @user @user Man b***h am not gonna do a damn thing; if u don't like what I got to say, then unfollow me	4	10
4.	firefighter is a job for white people	2	8

6.2.8 Error analysis

Though NACL performs better than other baselines across different evaluations, it does commit mistakes.

Prediction errors. Table 6.8 lists a few examples along with their original and predicted hate intensity scores. NACL-HIP assigns a lesser intensity score to the second utterance – the hatred is conveyed by a single word '*qu**r*,' and most likely, our model does not assign adequate focus on it in the presence of the positive phrase 'happy birthday'. In the third example, the predicted high score is possible because of the presence of three words -'*bi*ch*', '*damn*', and '*unfollow*'. We hypothesise that our model learns to associate colour as a cue for hate, thus predicting a high hate intensity score for the last example. We also encounter some misclassifications in hateful span identification, as shown in Table 6.9. In span identification, NACL-HSI commits three kinds of mistakes – (a) removes a token from the predicted span; (b) includes additional tokens as part of the hateful span; and (c) classifies non-hate spans as hate. The first kind of mistake can be observed in Examples #1 and #2 in Table 6.9. The second and third types of errors are shown in Examples #3 and #4, respectively.

Generation errors. Finally, Table 6.10 lists a couple of examples considering the hate normalisation task. In the first example, NACL-HIR, the generated text has low adequacy and fluency. Similarly, in the second, we see that the generated post is relatively more fluent than the first. Still, the intensity value is on the higher side due

Table 6.9: Gold and predicted spans for NACL-HSI, with spans indicated in blue. The green text in the second column indicates mispredictions.

#	Gold Span	Predicted Span
1.	Why do girls follow me? All I do is make fun of them {for how dumb they are}$_{Span}$ { #WomenS**}$_{Span}$	Why do girls follow me? All I do is make fun of them for {how dumb they are }$_{Span}$ {#WomenS**}$_{Span}$
2.	{go f*** yourself you stupid ugly c***}$_{Span}$ @user	go f*** yourself {you stupid ugly c***}$_{Span}$ @user
3.	Blac Chyna is {straight trash and an abomination to women}$_{Span}$ everywhere . Don't @ me cause I don't care. {H** responsibly , b***es .}$_{Span}$	Blac Chyna is {straight trash and an abomination to women}$_{Span}$ everywhere . {Dont @ me cause don't care .}$_{Span}$ {H** responsibly , b***es .}$_{Span}$
4.	okay {b***h, f**k off}$_{Span}$ its not your business {fall in a hole and get a**l f***d by satans horn! B***H}$_{Span}$ @user	{okay b***h}$_{Span}$, f**k off its not your business {fall in a hole and get a**l f***d by satans horn! B***H @user }$_{Span}$

Table 6.10: Erroneous generations of NACL-HIR.

#	Type	Example
1.	Original	*#LateNightThoughts how many Congressman {d***s did women s**k} to finally gain voting rights*
	Reference	*#LateNightThoughts how many Congressman {did women approach} to finally gain voting rights*
	Generated	*#LateNightThoughts how many Congressman {did women s**k} to finally gain voting rights*
2.	Original	*{S**s are half breed trash}. No {filthy native} should be allowed to speak to any European.*
	Reference	*No {native} should be allowed to speak to any European.*
	Generated	*{Mexicans are t**h}. No {disgusting native} should be allowed to speak to any person.*

Table 6.11: The left part of the table shows an example of human evaluation where we give the generated text to human annotators to evaluate the output of each competing model on three parameters – intensity, fluency, and adequacy. The results are an average of all the responses received for this example. The right part of the table shows the average scores of all three parameters across all responses (i.e. overall system averages). Intensity ranges from $1 - 10$, Fluency and Adequacy from $1 - 5$.

					Overall Average		
Model	**Tweet**	**Int ↓**	**Flu ↑**	**Adq ↑**	**Int ↓**	**Flu ↑**	**Adq ↑**
Original	*@user B*** you shouldn't be putting it out there c*** nu**et*	6	-	-	7.7	-	-
Dictionary	*@user b*** you shouldn't be putting it out*	5.72	3.45	3.45	4.38	1.08	3.09
FGST	*you shouldn't be putting it out there woman nu**et*	4.72	3.82	3.82	5.36	4.08	3.61
Bias Neut	*@user you shouldn't be putting it*	3.28	2.45	2.64	6.54	3.15	3.6
NACL-HSR	*@user you shouldn't be putting it out there*	2.36	4.54	3.90	**4.08**	**4.38**	**4.16**

to the inclusion of the hallucinated phrases '*Mexicans are t**h*' and '*disgusting*.' We argue that the problem of partial normalisation can be effectively addressed with more volume and variety of training samples, as well as explicitly controlling for the length of the statement.

6.2.9 Human evaluation as proxy for online users

In order to check the viability of the overall system, we perform a human evaluation and assess the generated text from NACL-HIR. We prepare a questionnaire to measure the quality of the generated texts. For a subset of hateful samples from our dataset, the human annotators are provided with four outputs corresponding to four high-performing hate normalisation systems, i.e., Dictionary-based, Bias Neutralisation, FGST, and NACL. To reduce bias, we anonymise the systems, randomly shuffle the outputs, and label them as A, B, C, and D.

Evaluator recruitment. Akin to the previous chapter, we recruit 20 human evaluators aged 18+ who have experience in using social media and work in CSS and NLP. The evaluation is voluntary, with no monetary compensation. The annotators provided background about the task, hate speech, and vulnerable groups. Having shared the information, those willing to participate are provided access to a Google form to evaluate 15 random samples of the test set. Among 20 annotators, 10 are male, and 10 are female. The age of all the annotators ranged between 25-40 years. All of them are social media savvy and understand the placement of a proactive mitigation tool in the social interaction pipeline.

Evaluation guideline. After reading the original sample and the normalised counterparts, you are to provide input on the following three dimensions – *hate intensity*, *adequacy*, and *fluency* (Bhattacharyya, 2015). Adequacy (the higher, the better) measures the semantic perseverance in the generated text. In contrast, fluency (the higher, the better) refers to the linguistic smoothness in the target language.

- Intensity. The annotators assign a hateful intensity score to each generated output on a scale of [1,10], 10 being the highest.
- Fluency. To understand how well constructed and readable the generated text is, the annotators scored each generated text on a range of [1,5], 5 being the highest fluency.
- Adequacy. Additionally, to provide an idea of whether the desired meaning can be interpreted from the output, the annotators were asked to score each generated text on a range of [1,5], 5 being the highest. Since our task aims to perform normalisation and not the conversion of hate to non-hate, if a sentence changes the sample's polarity, then that can also be taken as a negative case from our intended perspective. The annotators are informed beforehand that a normalised sample with its polarity reversed would have a minimum (1) adequacy even if it is fluent.

For reference, the intensity of the original sample is also provided. The fluency and adequacy of the original sample are considered the highest by default.

Observations. We present the average scores for one sample and the overall average scores across all samples in Table 6.11. On average, `NACL`-HIR outperforms others, with hate intensity of 4.08, fluency of 4.38, and adequacy of 4.16.

Table 6.12: Human evaluation of `NACL` with external samples.

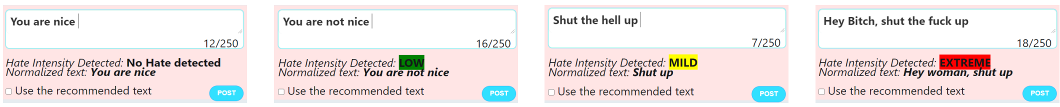
	Reddit			Gab			Facebook		
	Int $\downarrow$	**Flu** $\uparrow$	**Adq** $\uparrow$	**Int** $\downarrow$	**Flu** $\uparrow$	**Adq** $\uparrow$	**Int** $\downarrow$	**Flu** $\uparrow$	**Adq** $\uparrow$
FGST	5.12	2.26	1.63	5.70	2.32	1.43	6.08	2.9	1.45
Bias	3.28	2.02	1.01	3.89	1.82	1.02	6.47	2.41	1.06
NACL	**3.25**	**3.8**	**1.92**	**3.29**	**4.25**	**2.71**	**3.2**	**4.05**	**2.6**

Near real-time assessment

So far, via automatic and manual evaluations, we have looked at test samples drawn from identical distributions to that of the train set. However, given the proactive nature of the task, we need to assess how the `NACL` will work with out-of-distribution samples. We perform two in-the-wild analyses, one where we take random samples from the Web but from datasets different from the ones in the train set. The second analysis invites human participants to add random manual sentences and assess them in real-time. Both these external evaluations help test the robustness of the model against a host of inputs possible in the real world.

Platform independent evaluation. In this setup, we test `NACL` on 100 randomly sampled hateful posts from three distinct platforms – selected from Reddit, Gab and Facebook, obtained from Qian *et al.* (2019) (for first two) and Chung *et al.* (2019). For these systems, we do not have gold predictions. Instead, we extend the human evaluation for out-of-distribution analysis of `NACL`. We employ the same set of annotators and annotation process mentioned previously to evaluate the quality of `NACL` and the two best baselines in terms of intensity, adequacy, and fluency. Table 6.12 shows that `NACL` performs well compared to others across datasets. It is able to bring down the hate intensity to $\approx$ 3.25 (out of 10) while keeping the fluency very high with an average score of ≈ 4.03 (out of 5). Our cross-platform analysis further supports the universal nature of the task, allowing the system to be a standard plugin for multiple platforms.

Interactive analysis. Further, given the proactive nature of the problem we want to solve, it is imperative to assess how the system behaves in an interactive setup. Thus, we evaluate the tool in the wild by asking 25 participants (5 additional recruited apart from the initial 20) to write random hateful content on their own and rate the tool's output. We extend the comparative assessment by developing an internal =side-by-side interface consisting of NACL and the Bias Neutralisation baseline. The interface shows the outputs of NACL and the baseline for a given input in real time. Each participant is asked to input hateful content, assign an original intensity score, and subsequently evaluate the tools' outputs considering fluency, adequacy, and intensity. The normalisation methods are anonymised for the participants. In total, we obtain 100 input samples, with an average normalised intensity score of 3.24. Similar to earlier observations, evaluators notice that NACL results in more fluent sentences with a higher reduction in intensity. This interactive evaluation further supplements that NACL is not restricted to our dataset.



(a) No-hate (b) Low-hate (c) Mild-hate (d) Extreme-hate

Figure 6.4: Snapshots of the web extension for four scenarios. NACL generates normalised text only if $\phi_t > \tau$. The web framework detects hate as the user types in, and if any, $\phi_t > \tau$., it shows the level of hate that is detected in the current text and then recommends the normalised text to the user.

6.2.10 Towards deployable systems

We end the discussion by pointing out some engineering design decisions, keeping in mind the real-time interactive setup we propose with NACL.

Proposed tool. Finally, we propose a web interface that can easily be made to work in cross-platform settings, as corroborated by in-the-wild evaluations. The web service analyses the composed text on the go, reports the intensity of hate, and upon finding the text hateful ($\phi > \tau$), it suggests a normalised text as an alternative. The interface is developed via Flask[7] and works in an auto-complete fashion. In Figure 6.4, we show the snapshots of the tool for four input scenarios – *no-hate* ($\phi \sim 0$), *low-hate*($\phi \leq 5$), *mild-hate* ($\phi \leq 7$), and *extreme-hate* ($\phi > 7$).

> The interactive nature of the proposed interface is akin to an automatic spell checker, which analyses the content on the fly and quickly provides alternate suggestions to the users as they type.

Engineering tradeoffs. As stated at the beginning of this chapter, we aim to find a solution that not only performs well for the task but can also be deployed. Interestingly, from our experimentation, we observe that a simpler architecture produces a performance comparable to that of a complex solution. Subsequently, we make a tradeoff between

[7] https://palletsprojects.com/p/flask/

efficiency and complexity at each modelling stage. For instance, adding additional BiLSTM layers for HIP did not significantly boost the performance. Additionally, with HSI, more straightforward embedding solutions like Glove and ELMO perform comparably to BERT. We think the main reason for BERT's low performance is that the BERT tokeniser splits the words into subtokens, which causes the spans to be distributed unequally across the subtokenized words. This skews the target values that belong to one of the tags ($o_i \in (B, I, O)$), resulting in the model overfitting. For both HSI and HIR, we make the obvious decision to use the distilled versions of the larger BERT and BART models, further reducing the number of parameters to train and deploy. In the future, we would like to explore developing a single end-to-end trained system further to reduce the number of training and deployment parameters. Another direction would be the integration of prompting setups that have an ineligible training footprint and can be accessed via external APIs.

However, for such systems to be eventually deployable, they need to be trained and tested on a substantially large dataset and include participation from more human evaluators. As hatefulness is subjective, it can never be fully and explicitly encoded. Still, by engaging in a more diverse set of data points and human evaluators, more perspectives can be incorporated to bring the system to legally and socially acceptable standards.

6.2.11 Limitations and future work

- **Lack of real-world field experiments.** While we perform an array of intrinsic, extrinsic and human evaluations on `NACL`, accompanied by cross-platform and interactive assessments with competing baselines, the system has not been tested at scale. In the future, we aim to develop not just a proof of concept but also web plugins. Further, rigorous testing against adversarial setups, noisy texts (emojis, URLs), innuendos, etc, is needed to gauge the effectiveness of the system.
- **Dependency on manual annotations.** Currently, `HateNorm` only contains $3k$ parallel samples owing to heavy manual curation. This reduces coverage of hateful behaviour in the datasets. It also restricts the usage of more sophisticated generative models that require more examples to train. One way to overcome this can be the use of in-context learning examples to generate additional training samples and perform n-shot normalisation (Agarwal *et al.*, 2023).
- **Extending beyond explicit English spans.** In the current work, we skip over the implicit hateful samples due to the absence of explicit hate spans. In the future, we would like to put in rigorous effort to handle such cases and increase the size of the dataset. Implicit samples will already be on the low-hate intensity spectrum. Here, the intensity prediction may require the support of systems like `FiADD` and `Tox-BART` to map the underlying explicit hatefulness. However, they cannot be employed to generate the normalised versions of implicit hate. Additionally, it would be interesting to see how `NACL` can be extended to non-English texts, as hateful phrases in one language may not be offensive in another. Moving beyond spans to paraphrasing at a sentence level while maintaining intended semantics would be the direction.

6.3 Chapter conclusion

Acting as a bridge in the grey area between free speech and content moderation, we propose an alternative proactive solution via *hate speech normalisation*. The task is accompanied by the novel manually curated `HateNorm` dataset and the neural hate speech normaliser model `NACL` trained on `HateNorm`. An exhaustive evaluation system establishes not only the efficacy of the model under different settings but also highlights the cross-platform viability. In future, we hope to combine techniques from hate speech detection to strengthen the intensity prediction system and include `FiADD` based systems to extend to implicit samples as well.

Role of proactive mitigation in current social media. By conducting an in-house human evaluation with 25 participants and 100 in the wild samples generated by the participants, this chapter offers a PLM-driven interactive tool to aid in the proactive mitigation of hate. We hope that the proof of concept in this paper, as well as contemporary literature, will motivate practitioners to shine a light on proactive mitigation tools that can nudge less offensive and eventually non-offensive behaviour from their users. With social media platforms moving away from reactive content moderation and reducing their financial support for the same, proactive sensitisation of people offline as well as online becomes even more critical [8]. Tools such as ours can then be developed as third-party plugins that are not associated with any social media platform but are compatible with a range of them.

Overdependency on human annotations. A significant drawback of `HateNorm`, as well as datasets proposed in previous chapters, is the over-dependency on human annotation and evaluation, which reduces the number as well as the diversity of samples that can be curated. While human-in-the-loop for content moderation is irreplaceable, the existing systems can undoubtedly benefit from the pseudo-annotations that LLMs are capable of, especially for English text. They can help in generating diverse samples that can be augmented in existing datasets, which humans can easily verify. In our final chapter, we endeavour to examine the same.

[8] https://www.forus-international.org/en/custom-page-detail/121638-social-media-changes-and-content-moderation-implications-for-civil-society-organisations

Part IV

Human-LLM Alignment

CHAPTER 7

Incorporating Contextual Cues for Human-LLM Alignment

“Ironically, I believe Picasso was right. I believe we could paint a better world if we learned to see it from all perspectives, as many perspectives as we possibly could.”
- Hannah Gadsby; Nanette

7.1 Chapter introduction

Parallel to the rising demand for content moderation, there is an increase in the popularity of LLMs to serve as proxies for human labourers for a range of NLP and social computing tasks. One advantage of this setup is that it can lead to a faster and cheaper increase in dataset sizes via pseudo/silver labelling. The human-LLM system can be beneficial for demographic groups that are underrepresented as well as overworked in hate speech moderation. *However, for subjective tasks such as hate annotation, where people perceive hate differently, the LLM’s ability to represent diverse groups is unclear.* At the intersection of annotation priming and zero-shot prompting, we conclude our research by establishing the role of context signals/cues in making LLM-based hate speech annotations reflect the preferences of a given vulnerable community or different cultural groups more closely.

Borrowing from the demographic identities of `CREHate` (Lee *et al.*, 2024), we assess if LLMs qualify as a proxy for annotators of a given identity or an annotation setup. The direct implication is obtaining cheaper and faster soft labels via LLMs for a demographic that is underrepresented among human annotators. We investigate the role of geographical cues, similar to the use of geolocation metadata by online platforms, in annotating hate. Apart from the post’s content, numerical metadata (Founta *et al.*, 2019), such as the number of likes, number of times a post has been flagged as offensive, etc, is often available internally to content moderators. We examine whether such statistics can better guide LLMs to reflect a community’s needs. The analysis is fascinating because our research has previously established that such metadata is helpful for contextual modelling of hatefulness (Masud *et al.*, 2021; Founta *et al.*, 2019; Kulkarni *et al.*, 2023). Our findings on two LLMs, five languages, and six datasets reveal that mimicking persona-based attributes does lead to annotation variability. Our work provides preliminary guidelines and highlights the nuances of applying LLMs in culturally sensitive cases such as content moderation. Cognizant of adversarial attacks on LLM prompting, the study balances both feature importance (corroborated by significance testing) and cautions against adversary features.

So far, in this thesis, we continue to employ standard metrics like BLEU and F1 against the generative capabilities of LMs. However, the full extent of the capabilities of

LLMs and generative AI (GenAI) more broadly is challenging to evaluate. We conclude the thesis by providing a remark on the future direction for evaluating GenAI outputs.

7.2 Role of prompt priming in LLMs-based annotations

7.2.1 Motivation

Research gap. Annotators from diverse backgrounds are necessary for bringing novel prospects to understanding toxicity. Variation in annotations with respect to demographics matters as it is reflective of their lived experiences. On the other hand, however, the background of the evaluators also contributes to annotation biases (Rottger *et al.*, 2022; Aroyo *et al.*, 2019*a*; Munn, 2020). One way to overcome human subjectivity is to augment human annotations with machine-generated labels. The use of LLMs for annotations in NLP tasks is still nascent (He *et al.*, 2023; Ostyakova *et al.*, 2023), and it is not foolproof. In a recently proposed `CREHate` dataset by Lee *et al.* (2024), annotators from five countries record their perception of hatefulness for the same post (in English). The authors also observed variation upon introducing the country name when prompting the LLM for similar annotations. Reproducing results from `CREHate`, we observe a gap between the human-LLM alignment of hatefulness. *This motivates us to investigate variations in LLM's output when primed with context, which, under a similar setting for humans, causes variability.*

Research questions. By probing LLMs under a zero-shot setting for the task of hate speech annotation, we examine:

RQ1: Can geographical cues improve human-LLM alignment?
RQ2: Can LLMs adequately internalise demographic personas?
RQ3: Are LLMs prone to numerical anchoring cues?

Contribution summary. Decision-making in the human (Slovic *et al.*, 1977) is influenced by prior and current information. However, for general-purpose LLMs to replicate the same effect, they must be externally cajoled/cued. Employing the above RQs, we investigate the difference between implicit (base prompt) and explicit (contextual prompt) nudging. In a two-party setup consisting of a human and a zero-shot LLM as annotators, we prompt the LLM with contextual information and observe the IAA variation (w.r.t the human label) in the LLM's output. We perform an exhaustive analysis spanning two LLMs, five languages, and six hate speech datasets. In turn, we investigate 86 prompts in English and 40 multilingual ones. Our study illustrates that explicit cues align better with human annotations (Masud *et al.*, 2024*c*). Yet, our findings caution against the blindsided use of LLMs for crowdsourcing in subjective tasks. In summary, this research can help practitioners gauge the LLM vs. manual effort needed in their content moderation pipeline.

Note on research scope. Firstly, the aim is not to establish SOTA for a given dataset. We do not engineer the highest-performing prompt for a specific dataset. Instead, our study helps provide a general assessment across the datasets. Secondly, finetuning on a single hate speech dataset does not necessarily transfer to out-of-distribution samples (Yin and Zubiaga, 2021). Lastly, the variability and nuances probed in this

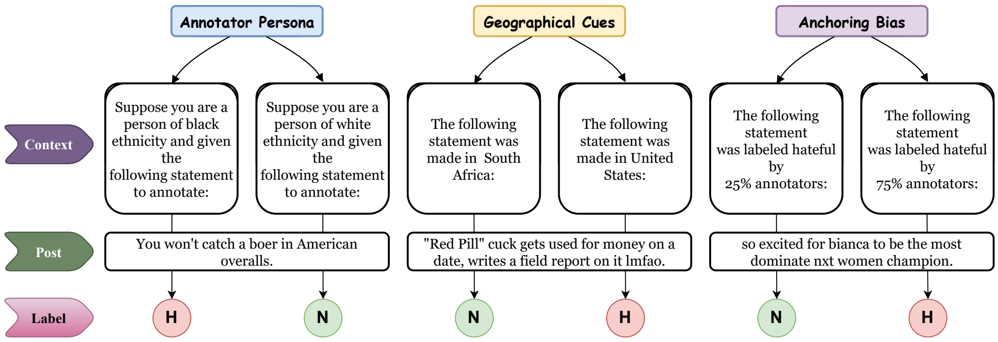


Figure 7.1: Overview of the research pipeline. An incoming post is prefixed with context to form the prompt for the LLM. The predicted label is then evaluated against the ground truth to examine variability arising from context.

study are not feasible to replicate under in-context and finetuned setups where multiple hyperparameters already impact the output. Therefore, in this study, we examine only the zero-shot prompt setting.

Table 7.1: Dataset statistics employed in this study. Here, Hate $\sim$ Hateful & Non-Hate $\sim$ Normal $\sim$ None.

Dataset (Language) (Reference)	# Samples in original dataset			# Samples used in RQs		
	# Hate	# Non-hate	Total	# Hate	# Non-hate	Total
`HateXplain` (En) (Mathew *et al.*, 2021)	4748	6251	10999	4748	6251	10999
`CREHate` (En) (Lee *et al.*, 2024)	709	871	1580	709	871	1580
`MLMA` (Ar) (Ousidhoum *et al.*, 2019)	460	915	1375	250	250	500
`MLMA` (Fr) (Ousidhoum *et al.*, 2019)	207	821	1028	207	293	500
`HASOC-2020` (De) (Mandl *et al.*, 2020)	146	1700	1846	146	354	500
`HASOC-2020` (Hi) (Mandl *et al.*, 2020)	234	2116	2350	234	266	500

7.2.2 Experimental setup

This section outlines the datasets, models, and evaluation metrics employed in the study.

Prompt format. The prompts are formatted as 'cue + post + query', where p_{base} = '*Statement: <POST>. Is the given statement hateful?*' forms the base/control prompt. RQ1 (Section 7.2.3) delves into the geographical nuances like the country or language of the post (e.g., *'The following statement was made in country x...'*) as a cue for regional context. RQ2 (Section 7.2.4) explores diverse demographic facets (e.g., *'Suppose you are a person of x ethnicity...'*) as a proxy for different human evaluator groups. Finally, RQ3 (Section 7.2.5) examines the variability introduced by numerical contexts, expressed as *'x% of individuals labelled this post as hateful.'*, as a proxy for anchoring bias in LLMs. Figure 7.1 outlines the integration of RQ-specific cues to p_{base}. Tables 7.2 and 7.2 provide the list of verbalised prompts under English and multilingual settings, respectively. We translate the cue and query into the respective language via Google Translate. Two linguistic experts verify a subset of translated prompts.

Datasets for evaluation. An overview of the datasets employed in this study is provided in Table 7.1. All these datasets are publicly available and contain a variety of toxicity labels. To remove the subjectivity of these umbrella terms, we categorise instances as *hate* when the label is either 'hate' or 'hateful,' and as *non-hate* when the label is 'normal,' 'none,' or 'non-hate.' We utilise the exact texts from the original

Figure 7.2: Random verbatim examples of multilingual prompts in the same language as the post (red) for the corresponding cues and queries (black) in English.

Post Language	Prompt in English language	Prompt in same language
Arabic	The following statement is in Arabic langauge and a non-muslim annotated this as non-hateful: صحيح كان كلاب ابن كلب مع شعبى بس كان صاكر دول بول البعيروكانت مملكة المنشار ترتجف خوفا اذا ظهر بخطاب بالتلفاز وال... @url. Is the given statement hateful? Answer in one word only.	العبارة التالية مكتوبة باللغة العربية وقد علق غير المسلم على ذلك بأنه غير مكروه: صحيح كان كلاب ابن كلب مع شعبى بس كان صاكر دول بول البعيروكانت مملكة المنشار ترتجف خوفا اذا ظهر بخطاب بالتلفاز وال... @url. هل البيان المعطى يكره؟ الإجابة بكلمة واحدة فقط.
French	The following statement is in French language and a french descent annotated this as hateful: @user @user l'islamisation en marche ! la pauvreté généralisée partout en france. Is the given statement hateful? Answer in one word only.	La déclaration suivante est en langue française et une descendance française l'a annotée comme haineuse: @user @user l'islamisation en marche ! la pauvreté généralisée partout en france. La déclaration faite est-elle haineuse? Répondez en un seul mot.
German	The following statement is in German language and a native german speaker annotated this as non-hateful: @neothemarius @niemehrgruen Nicht zu vergessen, einen moderaten ISlam... Is the given statement hateful? Answer in one word only.	Die folgende Aussage ist in deutscher Sprache und ein deutscher Muttersprachler hat dies als nicht hasserfüllt bezeichnet: @neothemarius @niemehrgruen Nicht zu vergessen, einen moderaten ISlam... Ist die gegebene Aussage hasserfüllt? Antworten Sie nur mit einem Wort.
Hindi	The following statement is in Hindi language and a person belongs to upper caste annotated this as hateful: RT @saitanBacha: @Kanatunga हाथ के कलर से दलित लग रहे हो 😅😅😂😂😂😂. Is the given statement hateful? Answer in one word only.	निम्नलिखित कथन हिंदी भाषा में है और ऊंची जाति के एक व्यक्ति ने इसे घृणित बताया है: RT @saitanBacha: @Kanatunga हाथ के कलर से दलित लग रहे हो 😅😅😂😂😂😂. क्या दिया गया कथन घृणास्पद है? केवल एक शब्द में उत्तर दीजिए।

datasets. We have not attempted to identify or remove any previously mentioned entities, such as the name of the target person in the given hate speech, etc. This decision ensures that the LLM receives the exact text for annotation as presented to the human annotator, maintaining consistency with the ground truth. For RQ1 and RQ2, we use all 1580 samples from `CREHate` (Lee *et al.*, 2024). Each sample in English is labelled as hateful or not by annotators from different nations (the United States, Australia, the United Kingdom, South Africa, and Singapore). For RQ3, we employ `HateXplain` (Mathew *et al.*, 2021), which contains English instances, encompassing three labels – toxic, hateful, and normal. We take samples with a majority label as hateful or normal, leading to $\approx 11k$ instances. We also investigate four datasets containing multilingual and code-mixed (with English) posts. Here, we use Arabic (Ar), French (Fr) (Ousidhoum *et al.*, 2019), Hindi (Hi), and German (De) (Mandl *et al.*, 2020).

Evaluation metrics. For p_{base}, we have the ground labels (majority voted gold labels). So, we employ a weighted-F1 score to compare the p_{base} outputs. Among the prompt variants in RQs, where we have a direct ground truth, we analyse the performance disparities via the Cohen-κ IAA (Cohen, 1960). It has been observed that IAA and F1 are positively correlated when the dataset is not skewed (Richie *et al.*, 2022). In our case, the skewness is controlled by having almost equal randomly sampled datapoints in both classes (as noted in Table 7.1). When choosing the IAA metric over the F1 score, our primary goal is not to emphasise high precision and recall. Instead, we aim to demonstrate how closely gold human annotations align with those generated by the LLM. This is why we employ IAA as our primary metric. We also use the predicted hate-label ratio (PHLR) as a metric. It represents the proportion of all samples labelled

Table 7.2: Prompts employed in this study for all settings under RQ1, RQ2, and RQ3. For `FlanT5` (all), $prompt \leftarrow prompt+$"Is the given statement hateful?" For `Llama3-8B`, `Mistral-7B`, `Zephr-7B` and `GPT-3.5`, $prompt \leftarrow prompt$ + "Is the given statement hateful? Answer in one word only." Here, $c \in$ *{United States, Australia, United Kingdom, South Africa, Singapore}*, $l \in$ *{Arabic, French, German, Hindi}*, $H/N \in$ *{hateful, non-hateful}*, $g \in$ *{male, female, non-binary}*, $e \in$ *{asian, black, hispanic, middle eastern, white}*, $p \in$ *{liberal, indepedent, conservative}*, $r \in$ *{christian, buddhism, islam, judaism, hinduism}*, $q \in$ *{high school, college, bachelor, master's, PhD}*, $B/V \in$ *{{muslim, non-muslim}, {french descent, mediterranean descent}, {native german, non-native}, {person belongs to upper caste, person belongs to lower caste}}*, $N/NN \in$ *{native, non-native}*, $z \in$ *{0%, 25%, 50%, 75%, 100%}*

RQ	**Prompt type**	**Prompt(s)**
Base	p_{base}	Statement: <POST>.
RQ_1	p_{con}	The following statement was made in <c>: <POST>.
	p_{lang}	The following statement was made in <l> language: <POST>.
RQ_2	$p_{trait}^{H/N}$	A <g> annotated the following statement as <H/N>: <POST>.
		A person of <e> ethnicity annotated the following statement as <H/N>: <POST>.
		A person of <p> political orientation annotated the following statement as <H/N>: <POST>.
		A person who follows <r> annotated the following statement as <H/N>: <POST>.
		An irreligious person annotated the following statement as <H/N>: <POST>.
		A person with an education level less than high school annotated the following statement as <H/N>: <POST>.
		A person with education level of <q> annotated the following statement as <H/N>: <POST>.
	p_{trait}^{A}	Suppose you are a <g> and given the following statement to annotate: <POST>.
		Suppose you are a person of <e> ethnicity and given the following statement to annotate: <POST>.
		Suppose you are a person of <p> political orientation and given the following statement to annotate: <POST>.
		Suppose you are a person who follows <r> and given the following statement to annotate: <POST>.
		Suppose you are an irreligious person and given the following statement to annotate: <POST>.
		Suppose you are a person with an education level less than high school and given the following statement to annotate: <POST>.
		Suppose you are a person with education level of <q> and given the following statement to annotate: <POST>.
	p_{trait}^{L*}	The following statement is in <l> langauge and a <B/V> annotated this as <H/N>: <POST>.
		The following statement is in <l> langauge and a <N/NN> speaker annotated this as <H/N>: <POST>.
RQ_3	$p_{vote}^{H/N}$	The following statement was labeled <H/N> by <z> annotators: <POST>.

as hate. It is calculated as the ratio of the total number of predicted hate labels to the total generated labels, excluding hallucinations and empty outputs.

Rectified scores for hallucination. We specify the prompt suffix 'answer in one word only.' Thus, any output not in the form of *'yes/no'*, *'hate/non-hate'*, or *'hateful/non-hateful'* can be considered as a 'hallucinated' label falling outside the range of the expected answers. We also perform a manual evaluation, and where the output could be salvaged, they are updated. In line with the existing literature (Lee *et al.*, 2024), after all filtering, we discard the outputs that still did not qualify as acceptable. Recording the number of hallucinations (discarded outputs), we introduce 'rectified F1/IAA scores' in

Table 7.3: Performance of LLMs when prompted with p_{base}. We report the number of samples in the data set used for prompting (# samples), the number of hallucinated outputs (# hal), and the rectified weighted F1/IAA. *close-sourced.

Model	# of parameters	HateXplain				CREHate			
		# Samples	# Hal	F1	IAA	# Samples	# Hal	F1	IAA
FlanT5-Small	60M	≈11k	2	0.412	0.000	≈1.5k	2	0.391	0.000
FlanT5-Base	250M	≈11k	85	0.649	0.341	≈1.5k	156	0.536	0.166
FlanT5-Large	780M	≈11k	4545	0.339	0.136	≈1.5k	572	0.411	0.187
FlanT5-XL	3B	≈11k	0	0.588	0.293	≈1.5k	4	0.638	0.292
Mistral	7B	≈11k	135	0.531	0.228	≈1.5k	198	0.568	0.303
Zephyr	7B	≈11k	3948	0.343	0.123	≈1.5k	560	0.323	0.102
Llama 3	8B	≈11k	1971	0.439	0.180	≈1.5k	679	0.357	0.150
FlanT5-XXL	11B	≈11k	0	0.731	0.476	≈1.5k	0	0.649	0.297
FlanT5-XXL	11B	500	0	0.738	0.487	500	0	0.649	0.297
GPT-3.5-Turbo*	>150B	500	0	0.780	0.576	500	2	0.758	0.517

Table 7.4: Performance of LLMs when prompted with p_{base} in Arabic and Hindi. We report the number of samples in the dataset used for prompting (#Samples), the number of hallucinated outputs (#Hal), and the rectified weighted-F1/IAA.

Dataset (Lang)	Model (# Params)	# Samples	# Hal	F1	IAA
MLMA (Ar)	Mistral-Ar (7B)	500	378	0.027	0.0
	GPT-3.5 (>150B)	500	222	0.359	0.140
HASOC-2020 (Hi)	Airavata (7B)	500	477	0.046	0.0
	GPT-3.5 (>150B)	500	131	0.257	0.018

Equation 7.1, where h and t are the hallucinated and total samples, respectively. Any mention of F1 and IAA in our study means 'rectified weighted-F1' and 'rectified IAA.'

$$score_{rectified} = (1 - \frac{h}{t}) \times score, score \in F1, IAA \tag{7.1}$$

LLMs for probing. We begin with FlanT5 (Chung *et al.*, 2022), Mistral (Jiang *et al.*, 2023*a*), Zephyr (Tunstall *et al.*, 2023), Llama 3 (Touvron *et al.*, 2023) and InstructGPT (Ouyang *et al.*, 2022) variant GPT-3.5-Turbo (hereby referred to as GPT-3.5). Performance metrics on all shortlisted LLMs (with the total number of parameters in that model) are captured in Table 7.3 for the p_{base}. From Table 7.3, we observe that even with p_{base}, most models generate noisy and ill-formatted outputs. To reduce the influence of noisy results, we select only FlanT5-XXL and GPT-3.5 to analyse more intricate setups in our RQs. Both these models portray negligible hallucinations and produce relatively higher F1 and IAA. To test FlanT5-XXL, we employ all the samples. For GPT-3.5, due to API rate limits, we test random 500 stratified samples per dataset.

We also examine language-specific LLMs to corroborate the role of zero-shot prompting setup of general-purpose LLM (GPT-3.5) with cultural context vs language-specific models. The results (Table 7.4) highlight that not just a demographic-based training corpus but also the manner of training and size of models impact multilingualism. Further, the results show that model instruction tuned on culture-specific corpus still falls short of aligning with the human annotators compared to GPT-3.5 under a multilingual setting.

Note on hate speech definition. Existing literature has observed that providing explicit definitions to LLMs may not lead to performance gain (Guo *et al.*, 2023; Roy *et al.*, 2023). We, thus, conduct our experiments building on the latent notions of hate speech in LLMs. For a sanity check, we prompt FlanT5-XXL and GPT-3.5 with

'Define: Hate Speech' and obtain following responses:

- FlanT5-XXL: *"Hate Speech is a form of speech intended to incite hatred or violence against a group of people."*
- GPT-3.5: *"Hate speech refers to any form of communication, whether spoken, written, or otherwise expressed, that promotes or encourages hatred, hostility, discrimination, or violence against individuals or groups based on attributes such as race, ethnicity, religion, nationality, gender, sexual orientation, disability, or other characteristics. It often involves derogatory language, stereotypes, or dehumanising rhetoric intended to intimidate, offend, or incite harm against the targeted individuals or communities. Hate speech is typically viewed as a violation of ethical principles. It may also be prohibited by law in many jurisdictions due to its potential to inflame tensions, provoke violence, and undermine social cohesion."*

Statistical testing. For RQ1 and RQ3, we perform the paired t-test and report the effect size. Meanwhile, for RQ2, we use ANOVA to capture the intra-demography disparity among the subclasses. The tests are performed on FlanT5-XXL.

Hardware and software(s). All experiments are conducted on an NVIDIA-A100 (80GB) GPU. Model weights are loaded from the HuggingFace pretrained models using float16 precision. For GPT-3.5, OpenAI's API is utilised. All models are employed in a zero-shot setting. The temperature variation experiment takes a total of 100 hours. The rest of the experiments collectively take $\approx$ 4-5 hours of GPU time. All statistical testing is run via the SciPy and NumPy libraries.

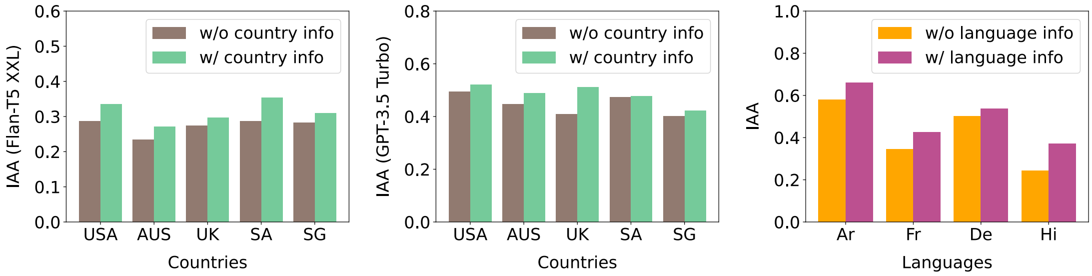
7.2.3 Do LLMs pick on geographical cues?

Hypothesis. Humans from different countries are prone to variability when flagging the same post as 'hateful' or 'non-hateful'(Lee *et al.*, 2024). Given that AI-assisted content moderation systems often have access to geolocation and language markers along with a post, we investigate whether *'akin to humans, do geographical cues influence LLM's predictions for an incoming hateful post?'*

Setup. Inspired by CREHate, we reproduce and extend their analysis on the language tag and multilingual datasets. As described in Section 7.2.2, the prompts are modified to suit our format. We compare the change in IAA when considering the human annotator of the respective country vs. p_{base} against the human annotator vs. $p_{con} + p_{base}$ or $p_{lang} + p_{base}$. Here, p_{con} = *'The following statement was made in <country>.'*, where $country$ = *{United States (USA), Australia (AUS), United Kingdom (UK), South Africa (SA), Singapore (SG)}*. The same set is used in CREHate as well. Meanwhile, p_{lang} = *'The following statement was made in <lang> language.'*, where $lang$ =*{Arabic (Ar), French (Fr), German (De), Hindi (Hi)}*. For analysing p_{con}, we employ CREHate for both FlanT5-XXL and GPT-3.5. For p_{lang}, we employ the HASOC-2020 (German & Hindi) and MLMA (Arabic & French) with GPT-3.5 English prompts. We run additional experiments with GPT-3.5 where the cues and query are in the same language as the post.

Findings. We discuss the results in two broad settings:

- **Country.** From Figure 7.3(b) with p_{con} + p_{base}, GPT-3.5 exhibits disparity in context about social constructs of the Global South (SA, SG) compared to



(a) `FlanT5-XXL` (b) `GPT-3.5` (c) `GPT-3.5`

Figure 7.3: (a-b) The IAA w.r.t human annotation for each country `FlanT5-XXL` and `GPT-3.5`, respectively, for English posts. (c) Captures each language's IAA w.r.t human labels via `GPT-3.5` with posts in the language and prompts in English. Here, without (w/o) is p_{base} and with (w/) is $p_{con}/p_{lang} + p_{base}$.

Table 7.5: Effect size to indicate the significance of including the country cue for `FlanT5-XXL` in English. *($p \leq 0.05$) and **($p \leq 0.001$) indicate whether the difference is significant.

Country	**ES F1**	**ES IAA**
United States	1.116*	1.165*
Australia	0.731*	0.729*
United Kingdom	0.723*	0.707*
South Africa	1.244*	1.495*
Singapore	0.782*	0.834*

higher alignment with so-called Western nations (USA, UK, AUS) (Zhou *et al.*, 2022; Li *et al.*, 2022; Lee *et al.*, 2024). Surprisingly, for `FlanT5-XXL` (Figure 7.3(a)), we observe an improvement in all countries. It is further corroborated by significance testing on `FlanT5-XXL`. From Table 7.5, comparing both F1 and IAA, we observe that adding the country cue leads to a significant change in the prompted output (as captured by the higher effect size (ES) as well as the p-values). Following the reference Figure 7.3(a), South Africa registers the highest impact. One hypothesis for the difference in p_{con} results can be the nature of training the LLM. With only instruction tuning, `FlanT5-XXL` develops country-specific bias implicitly from the training data; however, for `GPT-3.5` the implicit bias is augmented with explicit human feedback. It surely calls into question how LLM pretraining mechanisms impact the subjective (non-GLUE) downstream tasks such as hate speech detection (Roy *et al.*, 2023; Zhang *et al.*, 2025).

– **Language.** From Figure 7.3(c), we conjecture that p_{lang} nudges `GPT-3.5` to language-specific subspaces, leading to a visibly higher IAA with ground truth labels when we add p_{lang}. As we infer from Table 7.6, comparing both F1 and IAA, the delta increase in performance in p_{lang} + p_{base} vs p_{base} follows the same order of magnitude irrespective of whether the prompts are in English or the respective language. Despite the expected loss in performance (Jin *et al.*, 2024), the inclusion of p_{lang} leads to a better human-AI alignment.

Takeaways. The findings encourage incorporating geographical cues in the zero-shot prompt to ensure higher human-LLM alignment. Interestingly, even if the prompt is in the respective language, explicitly nudging helps. Our observations emphasise the fact that both corpus and training paradigms play a role in the geographical sensitivity

Table 7.6: Extension of Figure 7.3(c) for IAA comparison of the results without and with language cues for prompts in English and the respective language.

Language	Prompt in English		Prompt in same language	
	p_{base}	p_{lang}	p_{base}	p_{lang}
Arabic	0.580	0.660	0.140	0.305
French	0.344	0.425	0.272	0.356
German	0.502	0.537	0.412	0.423
Hindi	0.242	0.371	0.018	0.031

of the LLMs. As a quick fix, we suggest augmenting the language/country tags for an incoming post being prompted to improve alignment with people representative of a given region/geography. However, how the latent spaces are triggered at the mention of geographical cues is intractable from prompting alone (Zhou *et al.*, 2022). This becomes especially tricky for overlapping signals like 'Arabic,' 'Muslims,' and 'Islamophobia' (Figure 7.3(c)). It calls for more transparency in the LLM training to decode the geographical bias during training epochs.

> We observe that incorporating geographical cues leads to a significant increase in human-LLM agreement. As these metadata are readily available within a platform, including them in the prompt does not require additional manual effort.

7.2.4 Can LLMs mimic an annotator's persona?

Hypothesis. Socio-cultural experiences of humans colour their outlook about hate and cause variation in annotations (Sap *et al.*, 2019; Orlikowski *et al.*, 2023). However, without direct signals about mental state, we rely on demographic attributes as markers of human conditioning (Aroyo *et al.*, 2019*b*; Sap *et al.*, 2022) a.k.a *personas*. Meanwhile, LLMs only possess statistical socio-cultural experience. When employing LLMs for crowdsourcing, we need to assess *'do LLMs' emulation* of demographics at a fine-grained level, impact annotations?' We further hypothesise that differences in a vulnerable group's projection can lead to variability in the hate perceptiveness of LLMs. We consider a vulnerable group as those who have historically been mistreated based on identity (Kulkarni *et al.*, 2023).

Setup. Borrowing the annotator demographic list from `CREHate`, we define our persona attributes in terms of *D={Gender (D_g), Ethnicity (D_e), Political Orientation (D_p), Religion (D_r), Education Level (D_q)}*. Each demographic in $D_* \in D$ has further sub-classes as outlined in Table 7.7. For each D_*, we augment the prefix p_{trait} ='*A person who is <D*>, annotated the following statement as <H/N>.*' Operated via the <H/N> tag, we run two persona variants, calling the annotated statement either hateful p^H_{trait} or non-hateful p^N_{trait}. We also examine a third variant p^A_{trait} with the prompt p^A_{trait} ='*Suppose you are a person who is <D> and given the following statement to annotate.*' The 3 personas prompts ($p^*_{trait} + p_{base}$) are examined via the `CREHate` dataset on `FlanT5-XXL` and `GPT-3.5`.

Intrigued by the success of the language tag in RQ1 (Section 7.2.3), we deep dive into persona traits that closely represent each language's demographic. We start with

Wikipedia to get a general sense of the demographics of each geography. From there, we narrow down the most prominent demographics of the nation. We further use census and news articles and consult social experts about each geography before narrowing down the vulnerable minority based on the hypothesis that minority groups receive more hate than the majority. Here, $\forall l \in lang =$ *{Arabic, French, German, Hindi}*, we introduce prompts $p^{L*}_{trait} =$ *'The following statement is in <l> language, and a <I> annotated this as <H/N>.'* In the prompt, I enlists a base/majoritarian persona vs. a vulnerable/minority persona of the respective geography examined on `GPT-3.5`for:

- Arabic, $p^{L_{Ar}}_{trait} \in$ {*Muslim/Non-mulsim*}
- French, $p^{L_{Fr}}_{trait} \in$ {*French/Mediterranean descent*}
- German, $p^{L_{De}}_{trait} \in$ {*Native/Non-native German speaker*}
- Hindi, $p^{L_{Hi}}_{trait} \in$ {*Upper/Lower caste*}

We examine the hateful (H) and non-hateful (N) queries here as well, both with English and multilingual prompts. As none of the datasets provide demographic-specific labels, we rely on IAA between predicted and the majority-voted gold labels for our assessment. For $p^{H/N}_{trait}$, we provide in the prompt if the persona identifies a statement as hate/non-hate speech. Our objective here is to assess the role of these traits in persuading LLMs to increase/decrease the number of hate labels in their responses. We thus utilise the PHLR metric in addition to IAA.

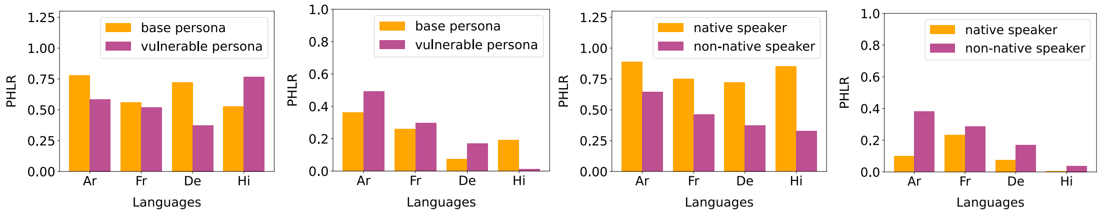
Findings. We discuss the results broadly for English and multilingual datasets.

– **English.** Table 7.7 corroborates that nudging the model to assume a persona (p^{A}_{trait}) is different from presenting the LLMs with a persona ($p^{H/N}_{trait}$). Further, it is evident from PHLR that LLMs are more sensitive towards some demographic subclasses than others. Significance testing corroborates the same. From Table 7.10, we observe that different modes of the persona (p^H, p^N, p^A) are impacted by varying sub-classes. Interestingly, the subgroups within 'Religion' show considerable variation for the three persona prompts. Under the 'Gender' demographic for `GPT-3.5`, the percentage of hate labels is higher for 'Non-binary' than 'Males' (for p^{H}_{trait}). It aligns with the former being a more susceptible subclass of gender in the real world. Meanwhile, for `FlanT5-XXL`, the presence of the non-hate tag p^{N}_{trait} dominates the demographic information in the context. However, the opposite is not valid for p^{H}_{trait}. Despite the ground labels associated with samples being balanced across classes, p^{N}_{trait} for `FlanT5-XXL` still predicts the majority of the labels as 'non-hate.'

– **Multilingual.** From Figure 7.4(a-b), we hypothesise that explicitly known vulnerabilities like Islamophobia (Arabic + Muslim) and Casteism (Hindi + Lower caste) are better captured by the LLM than ethnicity, leading to higher sensitivity of these pairs towards hate when prompted in English. While the patterns persist under multilingual prompting for the rest of the languages, the lack of variation in p^{H}_{trait} Hindi is puzzling (Table 7.8). As the highest gap in p^{*}_{trait} English is observed for German, we deep dive into this prompt (native speaker vs non-native speaker) and repeat this for all languages under English and multilingual prompting. From Figure 7.4(c-d) and Table 7.9, one can conjecture that when a native-speaking persona considers a post as hateful, the model may be contextualising the higher acuity of the speaker to understand the geographical context. From Tables 7.8 and 7.9, we again observe the same pattern as in English, thereby showcasing that

Table 7.7: IAA and PHLR w.r.t majority voted gold label in `CREHate`, for `FlanT5-XXL` and `GPT-3.5`. The demographic attributes (D) are compared under the hateful p^{H}_{trait}, non-hateful p^{N}_{trait}, and assumed persona p^{A}_{trait} settings. $\forall D_{*} \in D$, we combine the p^{*}_{trait} + p_{base}.

Annotator demographics	Sub-classes	Flan-T5-XXL						GPT-3.5					
		p^{H}_{trait}		p^{N}_{trait}		p^{A}_{trait}		p^{H}_{trait}		p^{N}_{trait}		p^{A}_{trait}	
		IAA	PHLR	IAA	PHLR	IAA	PHLR	IAA	PHLR	IAA	PHLR	IAA	PHLR
Gender	Male	0.42	0.53	0.00	0.00	0.31	0.29	0.40	0.70	0.55	0.44	0.57	0.46
	Female	0.42	0.53	0.00	0.00	0.33	0.31	0.39	0.72	0.46	0.35	0.52	0.54
	Non-binary	0.42	0.42	0.01	0.01	0.32	0.29	0.31	0.77	0.45	0.38	0.53	0.58
Ethnicity	Asian	0.46	0.56	0.03	0.02	0.33	0.23	0.37	0.75	0.55	0.51	0.51	0.59
	Black	0.43	0.61	0.03	0.01	0.33	0.23	0.37	0.74	0.54	0.51	0.50	0.64
	Hispanic	0.45	0.56	0.03	0.01	0.36	0.24	0.39	0.71	0.56	0.49	0.51	0.62
	Middle Eastern	0.46	0.52	0.03	0.01	0.29	0.19	0.40	0.70	0.54	0.54	0.49	0.64
	White	0.46	0.54	0.03	0.01	0.36	0.24	0.40	0.69	0.51	0.57	0.52	0.56
Political orientation	Liberal	0.42	0.52	0.01	0.01	0.32	0.29	0.49	0.63	0.54	0.53	0.61	0.53
	Indepedent	0.43	0.58	0.00	0.00	0.32	0.29	0.46	0.65	0.58	0.48	0.54	0.50
	Conservative	0.44	0.53	0.00	0.00	0.31	0.26	0.49	0.63	0.53	0.56	0.48	0.47
Religion	Christian	0.39	0.53	0.03	0.03	0.33	0.31	0.44	0.68	0.53	0.58	0.53	0.48
	Buddhism	0.41	0.54	0.03	0.03	0.32	0.30	0.45	0.66	0.52	0.59	0.51	0.45
	Islam	0.43	0.49	0.02	0.01	0.37	0.26	0.47	0.65	0.52	0.54	0.52	0.53
	Judaism	0.45	0.48	0.02	0.01	0.30	0.22	0.46	0.68	0.50	0.60	0.52	0.55
	Hinduism	0.42	0.49	0.04	0.03	0.32	0.24	0.48	0.65	0.52	0.53	0.51	0.49
	Irreligious	0.40	0.39	0.02	0.02	0.30	0.25	0.44	0.68	0.54	0.57	0.54	0.43
Education level	<High school	0.42	0.46	0.01	0.01	0.31	0.30	0.52	0.60	0.54	0.51	0.50	0.49
	High school	0.41	0.52	0.00	0.01	0.29	0.29	0.45	0.65	0.56	0.52	0.53	0.50
	College	0.42	0.52	0.00	0.01	0.29	0.29	0.44	0.67	0.53	0.44	0.52	0.46
	Bachelor	0.42	0.52	0.00	0.00	0.28	0.29	0.43	0.68	0.56	0.47	0.50	0.44
	Master's	0.42	0.52	0.00	0.00	0.28	0.29	0.41	0.69	0.54	0.45	0.48	0.46
	PhD	0.42	0.51	0.00	0.01	0.27	0.28	0.47	0.65	0.53	0.42	0.51	0.49



(a) `GPT-3.5` (H) (b) `GPT-3.5` (N) (c) `GPT-3.5` (H) (d) `GPT-3.5` (N)

Figure 7.4: PHLR from `GPT-3.5` comparing $p^{L_*}_{trait}$ + p_{base} for Arabic, French, German, and Hindi. (a) and (b) capture the base vs. vulnerable persona. (c) and (d) capture the native vs non-native speaker persona.

the role of context is significant irrespective of the language under consideration. Although we have seen a similar pattern in PHLR, there is a significant degradation in the observed metrics compared to English.

Takeaways. Given the sensitivity of LLMs toward the combination of persona and label, we urge practitioners not to rely on LLM as a substitute for demographic attributes mindlessly. Moreover, our work establishes that the manner of personification of demographic attributes can lead to variation in hate labelling, with assumed persona (p^{A}_{trait}) being closer to the inherent knowledge and biases an LLM possesses. Here, again, we caution practitioners looking to adopt LLMs for crowdsourcing to experiment with different framings of personas and identity traits. The persistence of results in English and multilingualism is an advantage for researchers working on multilingual datasets. It can allow faster assessment of the LLMs without requiring separate prompt engineering for both setups.

We establish that not only the demographic information but also the 'manner/format' of imbibing the persona is equally crucial for LLM prompting.

Table 7.8: PHLR for vulnerable persona cues. Extension of Figure 7.4(a-b).

Language	Majority or vulnerable	Prompt in English p^H	Prompt in English p^N	Prompt in same language p^H	Prompt in same language p^N
Arabic	Muslim	0.778	0.364	0.992	0.737
	Non-muslim	0.586	0.492	0.525	0.483
French	French descent	0.558	0.262	0.666	0.170
	Mediterranean descent	0.520	0.298	0.649	0.176
German	Native	0.724	0.076	0.566	0.152
	Non-native	0.374	0.170	0.248	0.248
Hindi	Upper caste	0.528	0.192	0.998	0.282
	Lower caste	0.768	0.014	0.998	0.094

Table 7.9: PHLR for native persona cues. Extension of Figure 7.4(c-d).

Language	Speaker	Prompt in English p^H	Prompt in English p^N	Prompt in same language p^H	Prompt in same language p^N
Arabic	Native	0.890	0.100	0.764	0.099
	Non-native	0.646	0.380	0.567	0.901
French	Native	0.752	0.232	0.916	0.130
	Non-native	0.462	0.286	0.702	0.106
German	Native	0.724	0.076	0.566	0.152
	Non-native	0.374	0.170	0.248	0.248
Hindi	Native	0.852	0.004	1.000	0.753
	Non-native	0.328	0.038	1.000	0.858

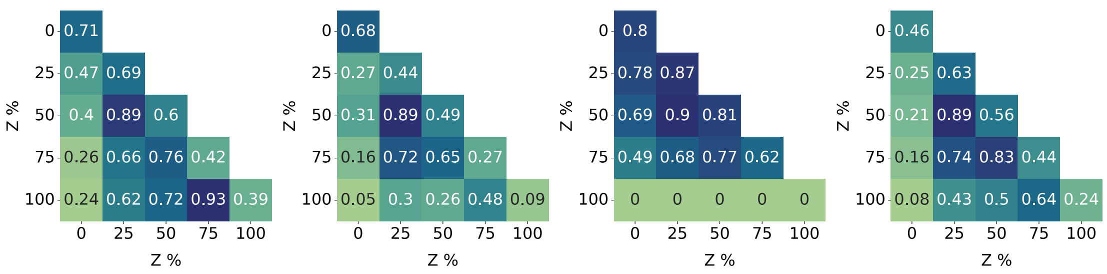
Table 7.10: The absolute difference between the minimum and maximum IAA for the respective persona, demographic, and LLM combination. *($p \leq 0.05$) and **($p \leq 0.001$) indicate if ANNOVA is significant among the sub-classes within a demographic.

Annotator demographics	p^H	p^N	p^A
Gender	0.010**	0.007	0.013*
Ethnicity	0.031**	0.027*	0.070
Political orientation	0.021*	0.011	0.007**
Religion	0.054**	0.021**	0.075**
Education level	0.013*	0.003	0.041

7.2.5 Are LLMs sensitive to anchoring bias?

Hypothesis. Anchoring bias occurs when humans rely too heavily on the anchor information (relevant or not) to influence their decision-making. While some anchors like language and geolocation, as established in Section 7.2.3, are helpful for LLM-based content moderation, the influence of numerical cues under the zero-shot setting is unknown. Numerical features/cues can be defined as aggregated real-world or simulated values representing countable metadata associated with the posts. It can range from the number of views, likes, and comments a post/user receives to the number of people who have in the past reported/flagged the post/user as hateful. The former set of metadata is publicly available and has been successfully employed by us in hate speech detection to improve modelling efficacy (Founta *et al.*, 2019; Kulkarni *et al.*, 2023). Meanwhile, the influence of previously flagged counts is not known, as such metrics are not available to the public. Here, using voting statistics as a proxy for the crowd's opinion about the post, we are motivated to examine whether '*made-up voting percentages in the prompt lead to manipulation in LLM's output?*'

Setup. The base prompt p_{base}, is prefixed with $p^{H/N}_{vote}$ = '*The following statement was labeled <H/N> by <z> annotators.*', where $z \in \{0\%, 25\%, 50\%, 75\%, 100\%\}$. The two variants $p^{H/N}_{vote}$ capture the hateful (H) or non-hateful voting (N) label. We represent z in the percentage to give a relative sense of majority voting. At $z = 50\%$, saying, 50%, annotators consider the post is hateful, and 50% say it is non-hateful, is not the same as p_{base}. We conduct this experiment with `HateXplain` on `FlanT5-XXL` and `GPT-3.5` with $p_{vote} + p_{base}$. The $z\%$ alludes to annotators in general and not a specific persona. Further, it should be noted that these percentages are not available as a part of `HateXplain` and are added by us to introduce the "anchoring" information.



(a) `FlanT5-XXL` (H) (b) `GPT-3.5` (H) (c) `FlanT5-XXL` (N) (d) `GPT-3.5` (N)

Figure 7.5: For the p^H_{vote} : (a) and (b) capture the IAA among various hateful voting percentages, i.e., $z\%$ for `FlanT5-XXL` and `GPT-3.5`, respectively. For p^N_{vote}: (c), (d) function analogously to (a), (b), respectively. Note: diagonals in heatmaps represent IAA between p_{base} and p_{vote} + p_{base}. Meanwhile, when x and y are different, it represents IAA b/w ($P_{vote} = x + P_{base}$) and ($P_{vote} = y + P_{base}$).

Findings. In line with existing evidence of majority labelling bias in few-shot learning (Zhao *et al.*, 2021), we also establish that LLMs are prone to labelling bias even under zero-shot settings if the context mentions voting percentages. Corroborated by significance testing (Table 7.11), we establish that while LLMs understand relative percentages, they are prone to emphasise this information over the post's content. From Table 7.11, for both p^H and p^N, we observe significant differences in performance among the various percentages in the prompt. From Figure 7.5(a-b) regarding p^H_{vote}, it is evident that alignment in hate labelling is more consistent when the percentages lie closer to each other and decrease as one moves away. Succinctly, $IAA_{ij} > IAA_{ik}$ if $|z_j - z_i| < |z_k - z_i|$. A similar pattern is observed for p^N_{vote} in Figure 7.5(c-d). We run additional experiments controlling for decoding temperature (Appendix I) and observe similar patterns.

Takeaways. As evident from our experiments, LLMs do not have a clear way of discovering noise from the informativeness in the numerically embedded prompt. As voting serves as a proxy of numerical metadata in the content moderation pipeline, it implies that corruption (accidental or intentional) can lead to instances of misclassification.

Table 7.11: The effect size for percentage pairs 'x' (key) and 'y' (value) indicates the significance of including the percentage cues. *($p \leq 0.05$) and **($p \leq 0.001$) indicate if the difference is significant or not.

Key	Value	p^H		p^N	
		ES F1	**ES IAA**	**ES F1**	**ES IAA**
0	25	2.24*	1.878*	6.379	4.47*
	50	2.732*	2.362*	1.25*	0.233
	75	3.805*	3.475*	4.342*	2.661*
	100	4.037*	3.583*	0.178	1.637**
25	50	1.53*	1.334*	4.416*	3.748*
	75	2.933*	2.626*	1.065*	1.412*
	100	3.23*	2.818*	3.064*	4.91*
50	75	2.516*	2.231*	3.701*	2.6*
	100	2.738*	2.36*	0.451*	1.559*
75	100	0.875*	0.773*	2.922*	3.493*

Pseudo-voting values influence predicted labels, highlighting both positive (community flagging) and negative (adversarial attacks) effects. Our advice to practitioners is to refrain from adding any numerical statistics in the prompt unless they are quality-checked and to mask such adversarial expressions written by the authors of the input samples.

Despite numerical/statistical features being used earlier by us and other researchers to contextualise LMs, (Founta *et al.*, 2019; Kulkarni *et al.*, 2023), the same cannot be directly extrapolated in LLM prompts for hate annotations. The results call into question the incorporation of numerical cues in zero-shot prompting in NLP.

7.2.6 Limitations and future work

- **Beyond zero-shot prompts.** The list of research questions and prompts analysed in this section is not all-encompassing. We hope our findings and primary analysis motivate future research. Given that the adoption of LLMs in subjective tasks such as hate speech annotations is still nascent (Huang *et al.*, 2023*b*; Roy *et al.*, 2023), in the future, we ought to expand the analysis to include more data, languages, and prompt formats. The role of in-context examples and toxicity definitions in driving personas is important to establish the extent of implicit personalisation in LLMs. *If LLMs are prone to majority bias, then our study opens up the questions around using few-shot/guideline examples for labelling hate.*
- **The scope of cultural markers.** It is essential to highlight that geographical or demographic cues are only partial representations of culture (Pawar *et al.*, 2024). Consequently, LLMs can also be a source of their implicit bias. While this is not a limitation of our work alone, quantifying the overlap and differences between culture, language, and stereotypes needs to be thoroughly examined before incorporating LLMs in subjective tasks (Manerba *et al.*, 2024; Naous *et al.*, 2024).
- **Accounting for annotator disagreements.** Owing to the lack of datasets with ground truth labels where multiple annotations are released as a part of datasets, performing one-to-one mapped IAA analysis between LLMs and demographic personas is challenging. Building consensus and modelling disagreement, (Fleisig *et al.*, 2023; Kralj Novak *et al.*, 2022) as a feature instead of majority-voted/flattened

consensus is a move towards better incorporation of diverse human experiences.

7.3 Towards a better evaluation of LLM hallucinations

7.3.1 Motivation

Research gaps. In the last section, we report rectified F1 (Section 7.2.2) to account for hallucinated and unfit label generations from the LLM. In cognitive science, the confabulation theory (Hecht-Nielsen, 2007) dictates that the brain develops false memories/impressions to tie missing/incomplete memories. Similarly, in GenAI, the output is confabulated owing to a lack of knowledge or inconsistency in learned patterns. Hallucination often manifests as unreal or unaligned content concerning the prompt (Guerreiro *et al.*, 2023). It also points to a lack of authenticity and dependability in the output (Menczer *et al.*, 2023; Dutta and Chakraborty, 2023). However, the discussion around how the nature of prompting impacts hallucinations and how to accommodate the variability in LLM output against standard evaluation metrics remains an open-ended question.

Research questions. In this discussion, we focus on how for LLMs and, more broadly, for GenAI:

RQ1: What changes in modelling and evaluation metrics are needed to reflect the variability and ingenuity in the output?

Contribution summary. For LLMs, a tradeoff exists between how much of the generation can be derived or transferred from preexisting concepts (learning during pretraining) and how much can be novel (adjusted via inference hyperparameters). In the first half of the discussion (Chakraborty and Masud, 2024), we raise the question of the extent to which the new entities can be introduced for the supposed hallucination to be considered creative and acceptable[1]. We conclude the discussion by enlisting mechanisms to accommodate the subjective evaluation of GenAI tools (Chakraborty and Masud, 2023).

7.3.2 Nature of prompting

As the output of an LLM depends on the input prompt, evaluating the extent of acceptable hallucination is underpinned by the nature of prompting (Figure 7.6) (Mukherjee and Chang, 2023). When given objective prompts, such as solving mathematical equations or listing the world's capitals, the output can be vetted against a source of truth. Meanwhile, subjective cues correspond to results with a vague conception of truth. When prompted to "write a poem" or "draw like Dürer," the outcomes can only be partially judged as art is perceptive. Then, there is a class of prompts like writing code snippets where the creativity in solutions can always be vetted against the correctness of the answer via test cases. Therefore, depending on the objectivity and novelty required by the task, we can decide the extent of acceptable hallucination.

[1] https://www.oreilly.com/radar/ai-hallucinations-a-provocation/

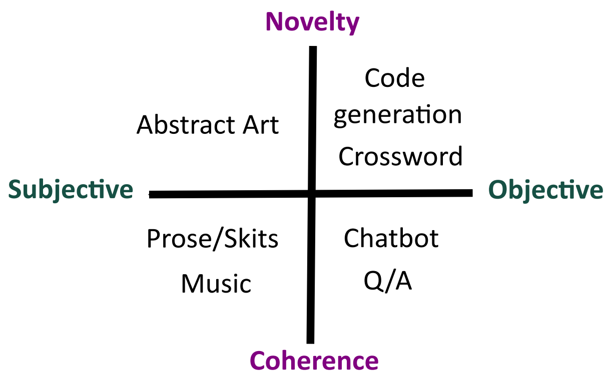


Figure 7.6: On the x-axis stands cues' subjective vs. objective nature. The more imaginative and emotional the prompting, the greater the difficulty in evaluating the creative output. On the y-axis stands the extent of coherence or factuality the model adheres to vs. the novelty it incorporates in its production. In each quadrant, we highlight some generative tasks that best describe the combination of prompting and hallucination that is possible for the task. The tasks can be evaluated under a closed solution set for objective prompts on the right side of the y-axis. Consequently, there are no exact evaluation criteria for the tasks on the left side of the y-axis.

7.3.3 Variability in output

Consider the case of personalised counterspeech generation via LLMs. If prompted to "quote Barack Obama to counter a racist tweet", there are three possible scenarios. In the first case, the model reproduces a quote verbatim from its parametric memory. In the second case, the result may be an inspired representation of Barack's speeches. In the last case, the model generates an outcome that is not in the style of Barack. Judging both individual outputs' faithfulness to the initial prompt as well as the impact of hallucination on creativity in generating a counter speech cannot be directly adjudged. Such use cases reinforce the dilemma of employing GenAI for subjective tasks. Even without hallucinations, the model is capable of inventive albeit predictable generations (Lee, 2023). It is akin to the evaluation metrics of fluency and specificity of the output text in natural language generation, which we adopted in Chapters 5 and 6 of the thesis.

Going back to our use case, there is an element of social and moral norms associated with quoting known figures in your speeches. Even if social information is present in pretraining data, it is one thing to replicate examples of socially acceptable behaviour and quite another to comprehend the circumstances in which those behaviours might be deemed inappropriate or objectionable. Guardrailing these behaviours, too, induces variability in the output. Note that here, we have spoken about the variability in output caused by input and latent variables. However, inference parameters like decoding temperature can also lead to variability.

7.3.4 Updating evaluation parameters

Theories of creativity have put humans (as individuals and as a society) at the centre of the creative process (Kaufman and Glăveanu, 2019; Glăveanu, 2013). However, these theories need revision to accommodate human-machine interactions and computer-aided generations. As GenAI exists only in relation to humans, does it mean that the outputs of GenAI will always be underpinned by human inventiveness and judged by human standards? The answer is not simple. A more fundamental challenge ahead of us would be to define the parameters under which GenAI output will be considered well-composed (Franceschelli and Musolesi, 2021) for varying degrees and difficulty of prompts and tasks. We propose that the research communities focus on building a dynamic test suite.

Secondly, contextual and implicit reasoning have only been weakly observed in GenAI via the chain-of-thought prompting setups (Wei *et al.*, 2022). A better understanding of social norms will help GenAI to better distinguish between originality and hallucination when it can be trained and evaluated on social norms. Being able to reason and pick up on implicit cues will improve the inventiveness of the output and help decode implied meanings in hate speech.

7.3.5 Limitations and future work

- **Synergising multiple sources.** Inevitably, an effort to understand the creativity of generative AI modes will also expand the horizons of human imagination and our perception of the world at large. The way forward is to encourage diverse researchers, artists, and users to contribute to developing such systems. Future designs of GenAI interfaces might incorporate a way that allows users to specify their desired tradeoff between coherence and novelty based on their goals.
- **Incorporating subjectivity in model training.** Another direction is to update the world's knowledge of GenAI systems dynamically (Ramapuram *et al.*, 2020). It should indirectly improve the system's ability to assimilate existing norms. One way is to extend the Reinforcement Learning from Human Feedback (Ge *et al.*, 2023) pipeline with more subjective prompts and social norms (Krishna *et al.*, 2022). An improved understanding of how the parameters gain knowledge (Yang *et al.*, 2023*a*) can help understand what gaps in knowledge lead to hallucination.

7.4 Chapter conclusion

As people from different socio-cultural ecosystems perceive toxicity differently, misalignment and variations in annotations show up when these attributes are prompted via LLMs (see Figure 7.7 for example). Moreover, in the current LLM literature, alignment and robustness do not have a singular non-overlapping definition, which makes the persona effects more difficult (Manzini *et al.*, 2024). Here, our current examination sits at the intersection of robustness in hate speech annotation as well as conceptualisation of the human-LLM alignment. The eventual goal is to provide LLM-based assistance that can complement (not substitute) human efforts in content moderation. To this end, over multiple RQs and prompt setups, we explore how well-suited LLMs are for assisting humans at different stages of the content moderation pipeline. Our results on demographic sensitivity, cultural priming, and anchoring bias are evident over multiple datasets, languages, and two LLMs. Our analysis reiterates using LLMs as an assistive system rather than replacing human moderators. We look forward to working on more datasets and exploring intersectional demographic attributes in the future. Interestingly,

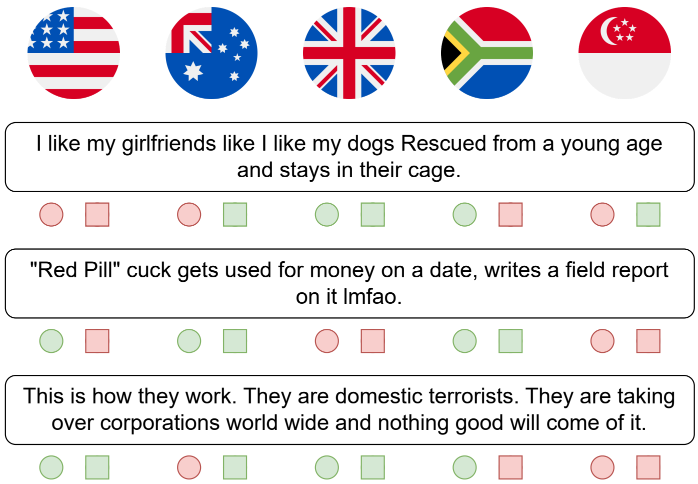


Figure 7.7: Annotations of hate/non-hate (red/green) for USA, Australia, Great Britain, South Africa, and Singapore, by a) annotators from respective countries (circle) and b) prompting `GPT-3.5` with "The following statement was made in <country>: <POST>. Is the given statement hateful?" (square). The posts and human labels are verbatim from `CREHate` (Lee *et al.*, 2024).

under multilingual prompting, we observe the (mis)alignment patterns persist more or less, albeit with an expected loss in performance (compared to English-only setups). The multilingual results are encouraging, allowing content moderators to work in the native language of the post without extraneous prompt engineering. The degradation in performance from English to multilingual (Jin *et al.*, 2024), however, calls for more investment in non-English training and evaluation of LLMs.

Not all contextual signals are helpful. It is equally important to remark on the fact that not all contextual cues are useful, especially numerical ones. Similar to our previous chapters, this chapter also reinstates the importance of contextual signals, now modelled as external cues in prompts; it also cautions against the blind use of these signals. Moreover, it is also imperative to point out that which contextual signal, whether textual, topographical, or statistical, is helping is a combined effect of how the signal is modelled, on what dataset the model is trained, and the desired outcome of the model.

LLMs for toxicity annotations. It is imperative to reiterate that hate speech is a human-centric phenomenon steeped in historical and cultural contexts. In this thesis and the majority of existing literature, a single label obtained via majority voting is the de facto practice, which does not account for the diversity arising from varied human experiences in relation to offline and online toxicity. As such, any computational attempt to flag it can only be assistive. Our experiments over three contextual settings indicate that LLM cannot outright substitute a demographic group in the annotation process or the content moderation pipeline. One way to understand the extent of hallucinations in LLMs is to compare the variation in results when prompts are varied for open-ended/subjective tasks. This comparison should also be extended to analysing the impact of hyperparameters like temperature.

Part V

Conclusion

CHAPTER 8

Thesis Conclusion

"We must not forget that when radium was discovered, no one knew that it would prove useful in hospitals. The work was one of pure science. This is proof that scientific work must not be considered from the point of view of direct usefulness. It must be done for itself, for the beauty of science, and then there is always the chance that a scientific discovery may become like the radium, a benefit for humanity."
- Marie Curie; Lecture at Vassar College

Social computing research aided by NLP and AI is emerging as the radium of our times. Sadly, the world is a constant reminder that toxicity cannot be contained by algorithms alone (Parker and Ruths, 2023). However, there is no long-term sustainable solution to combating harmful behaviours without *synergising humans and algorithmic moderation* (Kirtz and Talat, 2023; Chakraborty and Masud, 2022). It is now evident at a scale that human behaviour can be nudged for good or bad by the infusion of socially engineered technology (Rachmad, 2024; Celadin *et al.*, 2024). It is this characteristic of technology that is a ray of hope. It is with this hope that we approach the thesis on analysing, quantifying, and mitigating hateful text on the Web. Hateful behaviour, whether online or offline, stems as a byproduct of multiple latent social factors (Fischer *et al.*, 2018; Suler, 2004; Navarro, 2013). However, it is oppugning to determine "true" intent specifically in an online environment where non-verbal cues are pretty non-existent. Here, we propose *to determine the extent to which contextual signals can help assess the seemingly hateful text under consideration*. The need for employing contextual signals when modelling hate speech also comes from the fact that human annotators and moderators possess implicit and explicit dynamic world knowledge and experiences, which influence their understanding of hate speech (Rottger *et al.*, 2022; Aroyo *et al.*, 2019*a*; Munn, 2020). To initiate similar behaviour from computational models, we need external signals or contextual priming.

8.1 Contribution summary: *What did we manage to learn?*

Without anthropomorphising toxicity, we attempt to bridge the gap between human and computational modelling of hate speech. To this end, the thesis empirically establishes the role of contextual signals and *the takeaways can be summarised in terms of datasets, metrics, and modelling*:

T1: Due to the subjective and dynamic nature of hate speech, there is no standard benchmark method for collecting and annotating hate speech datasets. No single dataset can sufficiently encompass multiple tasks. We face this very challenge in this thesis, where moving from one aspect of research to another requires curating a newer dataset. For example, a mere translation of implicit hate samples from English to Hindi cannot capture the cultural connotations of Indian implicit hate speech. For this thesis, as well as NLP and CSS research, multiple non-standard datasets are not a bug but rather a feature of the hate speech literature (Vidgen and Derczynski, 2021; Fortuna *et al.*, 2020).

T2: In terms of standard evaluation metrics, contextual signals do improve the performance across hate speech-related tasks. The improvements are visible from feature-engineered logistic regressions to zero-shot GPT-3. Meanwhile, in terms of human evaluation, contextually finetuned PLMs outperform zero-shot LLMs in terms of specificity and adequacy towards hate speech-related tasks. Throughout our thesis, we observe that the so-called smaller language models, as well as finetuned systems, still hold value against a one-size-fits-all LLM replacement.

T3: Not all contextual information is helpful. The signals appropriate under one setup can prove counterintuitive in other use cases. It is not just the presence of contextual information but also the manner of infusing it that impacts the signal/noise ratio. Similarly, in terms of finetuning, the efficacy of hyperparameters is dataset-dependent, and this needs to be accounted for when optimising multiple datasets for hate speech detection.

Based on the above takeaways, two actionable insights in this research involve:

A1: Investigating the role of LLMs as pseudo-annotators rather than a means to an end to combat toxicity. These synthetic datasets so obtained can then be employed to finetune PLMs (Hartvigsen *et al.*, 2022).

A2: Formalising the baselines and evaluation metrics that can represent the efficacy of hate speech-related tasks better. While toxicity datasets cannot be standardised, the baseline systems and evaluation metrics can be used to a great extent. This includes reporting simple systems like logistic regression as well as bias-measuring metrics. Toxicity-specific model cards[1] and guardrails can developed and reported (Gehman *et al.*, 2020; Inan *et al.*, 2023).

8.2 Future Work: *How can we make content moderation better?*

Our thesis is a testimonial that human involvement in content moderation is pertinent (Scheuerman *et al.*, 2021), as there is no all-encompassing solution. As observed in Figure 8.1, there are numerous moving parts to analysing, monitoring, and combating hateful behaviour (Chakraborty and Masud, 2022), and this thesis only looked at a subset of the problems. Still, there are broader research directions within hate speech literature that need attention. Based on our understanding of handling multiple tasks and modelling setups, themes emerge around:

B1: Going back to the source, extensive work needs to be done to establish and report preexisting biases, stereotypes and toxic connotations in the pretraining datasets, and subsequently trained PLMs, LLMs and word embeddings employed for hate

[1] https://huggingface.co/blog/evaluating-llm-bias

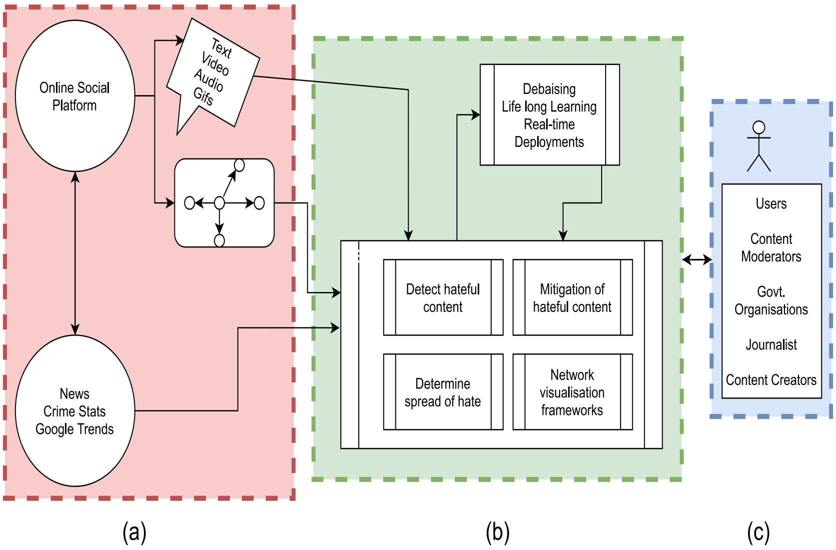


Figure 8.1: The framework for analysing and mitigating hate speech (**Chakraborty and Masud, 2022**) consists of the following: (a) Input signals. (b) Models for detecting hateful content and its spread. (c) Multiple stakeholders are at the receiving end of the framework. Their feedback and interactions directly impact the deployed systems.

speech tasks (including multilingual setups) (Nangia *et al.*, 2020; Ousidhoum *et al.*, 2021; Névéol *et al.*, 2022; Sahoo *et al.*, 2024; Longpre *et al.*, 2024).

B2: Study of toxic content and toxic behaviour of users needs to be examined in combination with their tendency to engage in other harmful behaviour. This includes exploring the relationship between offline and online hate speech (Lupu *et al.*, 2023; Williams *et al.*, 2019), hate speech and spread in misinformation, collusive attacks, and the long-term impact of exposure to hateful narratives.

B3: A/B testing of proactive mitigation techniques needs to be conducted at scale by independent auditors and platforms (Katsaros *et al.*, 2022; Ribeiro *et al.*, 2025). Under what circumstances does proactive mitigation fail, and how can it be effectively combined with explicit hate speech detection, which underpins proactive mitigation?

B4: The best solution for hate speech is counterspeech. However, right now, the hate speech detection and mitigation strategies operate in silos. We need more synergised pipelines that detect the type and target of hate speech, analyse the emotional and hateful intensity of the content, and then generate the most apt and specific counterspeech (Hangartner *et al.*, 2021; Mun *et al.*, 2024; Gupta *et al.*, 2023).

B5: It is also essential to highlight the gap between the social and psychological studies of harassment faced by vulnerable groups and their computational analysis by technical researchers and industrialists (Vendrell Ferran, 2024; Jhaver *et al.*, 2023). Again, the two systems operate in silos with little interaction between them. We need to incorporate and synergise social issues and their corresponding modelling.

B6: In our thesis, we did not cover the legal aspect of hate speech. In existing literature, the involvement of legal experts in the annotation or evaluation of samples curated from the web is missing. One way in which legal experts can be involved is by providing free text explanations of which legal clause the text violates, if any, and whether its hate intensity warrants any legal action. Using LLM prompts in the in-context information in the form of legal clauses can also be one way to perform the first level of filtering.

8.3 Limitations: *What are some open research questions?*

Despite the focus of the thesis being around hate speech-related tasks, our empirical observations lay out some interesting open-ended questions with broader implications for NLP and CSS. We list some of the long-term challenges below:

OQ1: The methods for extracting entities from knowledge bases are not just rudimentary but also task agnostic. While it guarantees scalability, it comes at the cost of performance on task-specific metrics.

OQ2: The information gain and variability in latent space introduced by domain-specific LMs or large-scale multilingual LMs need to be thoroughly examined. This can also unearth the best finetuning and prompting strategies for non-English, non-generalised PLMs and LLMs.

OQ3: How can research in the psychology of hate and nudging human behaviour be incorporated into the content moderation pipeline to reduce the promotion of harmful content? Further, how can in-person training and sensitisation programs be better integrated and monitored to reduce the generation of toxic content in the first place?

OQ4: Even for seemingly objective tasks like fake news detection, multiple versions and a spectrum of perspectives of the truth exist. Here, modelling annotator disagreements at the label or span level while being faithful to the latest information is difficult for both humans and LLMs. We also need to allow for multiple perspectives from a larger human annotator set to be incorporated. Further, employing LLMs to help annotate fine-grained tropes can help in understanding longitudinal narratives that promote harmful and misinformed content on the Web.

OQ5: Last but not least, we need to acknowledge that in the majority of the thesis, datasets have been evaluated by a maximum of 3 annotators (except for one instance of crowsourcing). To go from the current academic setup to a publicly accessible, large-scale system, we need more diverse datasets to be collected, annotated, and regularly tested.

In conclusion, we hope that our research into contextual priming to analyse multilingual and implicit hate better will inspire the research community to build an information-rich and human-centred content moderation system. We also hope that it encourages the development of proactive systems that can help reduce the visibility of toxic content both in offline and online spaces, allowing the Web to exist as a genuinely democratic and participatory space.

Part VI

Appendix

CHAPTER 9

Appendix

A Hate generation prediction

Problem statement. The problem of modelling hate generation can be formulated as assigning a probability to each user that signifies their likelihood to post a hateful tweet. With our hypothesis of hateful behaviour being a topic-dependent phenomenon, we formalise the problem as learning the parametric function, $f_1 : \mathbb{R}^d \rightarrow (0,1)$ such that:

$$P(u_i|\mathcal{T}) = f_1(\mathcal{S}^{\text{en}}, \mathcal{S}^{\text{ex}}, \mathcal{H}_{i,t}, \mathcal{T}|\theta_1) \quad (1)$$

where $\mathcal{T}$ is a given topic, t is the instance up to which we obtain the observable history of u_i, d is the dimensionality of the input feature space, and θ_1 is the set of learnable parameters. Though ideally, $P(u_i|\mathcal{T})$ should be dependent on $\mathcal{S}_i^P$ as well, the complete follower network for Twitter remains mostly unavailable due to account settings, privacy constraints, inefficient crawling, etc.

Table 1: List of model parameters used for predicting hate generation. *SVM-r* and *SVM-l* refer to Support Vector Machines with RBF and linear kernels, respectively. *LogReg*: Logistic Regression, *Dec-Tree*: Decision Tree.

Classifier	**Parameters**
LogReg	Random state=0
AdaBoost	Random State=1
SVM-r	Class Weight = 'Balanced'
SVM-l	Penalty= l2, Class Weight = 'Balanced'
Dec-Tree	Class Weight = 'Balanced', Max Depth = 5
XGBoost	eta=0.4, eval metric= 'logloss', learning rate=0.0001, objective= 'binary:logistic', reg alpha = 0.9

Feature engineering. To experiment on our hate generation prediction task, we use a total of $19,032$ tweets(which have at least 60 news mapping to it from the time of their posting) coming from $12,492$ users to construct the ground truth. With an $80:20$ train-test split, there are 611 hateful tweets among $15,225$ in the training data, whereas 129 out of $3,807$ are in the testing data. To deal with the severe class imbalance of the dataset, we use both upsampling of positive samples and downsampling of negative samples. With all the features discussed in Section 3.2.5, the full size of the feature vector is $3,645$. We experimented with all our setups with this complete set of features and dimensionality reduction techniques applied to it. We use Principal Component Analysis (PCA) with the number of components set to 50.

Topic (hashtag)-oriented feature. Additionally, we compute Doc2Vec (Le and Mikolov, 2014) representations of the tweets, along with the hashtags present in them as individual tokens. We then compute the average cosine similarity between the user's recent tweets and the word vector representation of the hashtag; this serves as the topical relatedness of the user towards the given hashtag.

Experimental setup. In the absence of external baselines for predicting hate generation probability due to the problem's novelty, we implement a total of six different classifiers using the Support Vector Machine (with linear and RBF kernel), Logistic Regression, Decision Tree, AdaBoost, and XGBoost. Parameter settings for each of these are reported in Table 1. For PCA, we conduct experiments by selecting K-best features ($K = 50$) using mutual information. We exercise extensive feature ablation to examine the relative importance of different feature sets. Among the six different algorithms we implement for this task, along with other sampling and feature reduction methods, we choose the best-performing model for this ablation study. Following Equation 1, we remove the feature sets representing $\mathcal{H}_{i,t}$, $\mathcal{S}^{\text{ex}}$, $\mathcal{S}^{\text{en}}$, and $\mathcal{T}$ in each trial and evaluate the performance.

Table 2: Performance of classifiers for the prediction of hate generation. *Proc.* signifies different feature selection and label sampling methods, where *DS*: downsampling of dominant class, *US*: upsampling of dominated class, *PCA*: feature dimensionality reduction using PCA, *top-K*: selecting top-K features with $K = 50$.

Model	**Proc.**	**Macro-F1**	**ACC**	**AUC**	**Model**	Proc.	Macro-F1	**ACC**	**AUC**	**Model**	**Proc**	**Macro-F1**	**ACC**	**AUC**
SVM linear	None	0.52	0.94	0.52	SVM rbf	None	0.55	0.88	0.61	LogReg	None	0.50	0.96	0.50
	DS	0.63	0.73	0.63		DS	0.62	0.70	0.64		DS	0.64	0.79	0.63
	US+DS	0.44	0.64	0.63		US+DS	0.46	0.69	0.66		US+DS	0.47	0.72	0.63
	PCA	0.55	0.90	0.59		PCA	0.48	0.71	**0.68**		PCA	0.49	0.97	0.50
	top-K	0.53	0.84	0.63		top-K	0.50	0.79	0.62		top-K	0.49	0.97	0.50
Dec-Tree	None	0.51	0.79	0.64	Ada-Boost	None	0.49	0.97	0.49	XGB	None	0.53	0.97	0.52
	DS	**0.65**	0.74	0.66		DS	0.62	0.77	0.61		DS	0.57	0.76	0.57
	US+DS	0.45	0.67	0.61		US+DS	0.44	0.63	**0.68**		US+DS	0.44	0.66	0.62
	PCA	0.46	0.68	0.65		PCA	0.50	0.97	0.50		PCA	0.51	0.96	0.51
	top-K	0.53	0.84	0.63		top-K	0.49	0.97	0.50		top-K	0.49	0.97	0.50

Performance evaluation. Table 2 presents the performances of all the models to predict the probability of a given user posting a hateful tweet using a given hashtag. It is evident from the results that all six models suffer from the sharp bias in data; without any class-specific sampling, they tend to lean towards the dominant class (non-hate in this case), resulting in a low macro-F1 and AUC compared to very high binary accuracy. SVM with RBF-kernel outperforms the rest when no upsampling or downsampling is done, with a macro-F1 of 0.55 (AUC 0.61).

Effects of sampling. Downsampling the dominant classes results in a substantial leap in the performance of all the models. The effect is almost uniform over all the classifiers except XGBoost. In terms of macro-F1, the Decision Tree sets the best performance altogether for this task as $\mathbf{0.65}$. However, the rest of the models lie in a very close range of 0.62-0.64 macro-F1. While the downsampling performance gains are explicitly evident, the effects of upsampling the dominated class are less intuitive. For all the models, upsampling deteriorates macro-F1 by a large extent, with values in the range of 0.44-0.47. However, the AUC scores improve by a significant margin for all the models with upsampling, except the Decision Tree. AdaBoost achieves the highest AUC of $\mathbf{0.68}$ with upsampling.

Dimensionality reduction of feature space. Our experiments with PCA and K-best feature selection by mutual information show a heterogeneous effect on different models. While only SVM with a linear kernel shows some improvement with PCA over the

Table 3: Feature ablation for Decision Tree with downsampling for predicting hate generation. At each trial, we remove features representing signals – $\mathcal{H}_{i,t}$ (*All* $\setminus$ *History*), $\mathcal{S}^{\text{ex}}$ (*All* $\setminus$ *Endogen*), $\mathcal{S}^{\text{en}}$ (*All* $\setminus$ *Exogen*), and $\mathcal{T}$ (*All* $\setminus$ *Topic*).

Features	**Macro-F1**	**ACC**	**AUC**
All	0.65	0.74	0.66
All $\setminus$ History	0.56	0.59	0.64
All $\setminus$ Endogen	0.61	0.68	0.64
All $\setminus$ Exogen	0.56	0.58	0.66
All $\setminus$ Topic	0.65	0.74	0.66

original feature set, the rest of the models observe considerable degradation of macro-F1. However, SVM with the RBF kernel achieves the best AUC of 0.68 with PCA. With top-K best features, the overall gain in performance is not very significant except for the Decision Tree. We also experimented with combinations of different sampling and feature reduction methods, but none of them achieved a substantial gain in performance.

Ablation analysis. We choose the Decision Tree with down-sampling of the dominant class as our best-performing model (in terms of macro-F1 score) and perform ablation analysis. Table 3 presents the performance of the model with each feature group removed in isolation, along with the entire model. Evidently, for predicting hate generation, *features representing exogenous signals and user activity history are most important.* Removal of the feature vector signifying trending hashtags, which represent the endogenous signal in our case, also worsens the performance to a significant degree.

B PLMs probing for hate detection

B.1 Impact of variations in random seed initialisation

Hypothesis. With no guarantee of attaining global minima via gradient descent, some seed initialisation of weights during pretraining could lead to better performance downstream. On the one hand, in a study over multiple seeded BERT (Sellam *et al.*, 2022), it was observed that the GLUE benchmark (Wang *et al.*, 2019) is susceptible to randomness in finetuning and especially pretraining seed strategy. Meanwhile, for auto-regressive models, it has been observed that the order of training samples during pretraining has a very low correlation with what the final model memorises (Biderman *et al.*, 2023). We hypothesise that hate detection should follow the former patterns.

Setup. We utilise the publicly available 25 different final checkpoints of BERT (Sellam *et al.*, 2022), each trained under the same architecture and hyperparameters but with different random weights (random seed) initialisations and shuffling of the training corpus. We randomly picked five pretrained checkpoints for our analysis. The seeds employed for selecting the five checkpoints will be referred to as the *pretraining seed set* ($ps = \{0, 5, 10, 15, 20\}$). To better capture the impact of pretraining weight randomisation, the PLM is frozen, and only the CH is trained. Further, to control for the randomness in the MLP layer, we use ms and run a differently-seeded (ms, ps) combination.

Table 4: Comparison of the minimum and maximum macro F1 obtained under varying seed combinations by each dataset. $S_{ms,ps}$ represents the combination of ms and ps. ES stands for effect size. ** and * indicate whether the difference in minimum and maximum macro F1 is significant by ≤ 0.05 and ≤ 0.001 p-value, respectively.

Dataset	Min F1	Max F1	ES
`Waseem`	$S_{451,0}$: 0.675	$S_{12,10}$: 0.731	0.446*
`Davidson`	$S_{451,0}$: 0.745	$S_{12,15}$: 0.792	0.582**
`Founta`	$S_{12,5}$: 0.872	$S_{127,20}$: 0.888	0.473**
`OLID`	$S_{451,0}$: 0.647	$S_{451,10}$: 0.731	0.287*
`HateXplain`	$S_{127,5}$: 0.630	$S_{451,10}$: 0.680	0.676**
`Dynabench`	$S_{451,15}$: 0.625	$S_{12,20}$: 0.660	0.724**
`Toxigen`	$S_{451,5}$: 0.767	$S_{127,10}$: 0.771	0.226

Findings. At the macro level, as outlined in Table 4, the performance appears to be significantly impacted by different seed (ms, ps) combinations. We perform a p-test on each dataset's overall minimum and maximum macro F1 seed pairs to establish the same. The difference in performance is significant for 5 out of 7 datasets with medium to high effect sizes. Similar to prior work (Sellam *et al.*, 2022), we look at the variability in performance when considering one set of seeds to be fixed. Table 5 and Table 6 provide a seed-wise breakdown comparing minimum and maximum macro F1 scores when employing the multiple-checkpoints BERT (Sellam *et al.*, 2022) model. In Table 5, the ms is constant, but ps varies and vice-versa in Table 6. It appears that keeping ms constant leads to more variability in performance than ps. It follows from the fact that in finetuning settings, the MLP layer initialised with ms is trainable, while the pretrained model initialised with ps may be fully or partially set to non-trainable (fully in our case). In this investigation, the machine-generated dataset (`Toxigen`) is the only one immune

Table 5: A comparison of minimum and maximum macro F1 is obtained when ms is constant but ps varies. ES stands for effect size. ** and * indicates whether the difference in minimum and maximum macro F1 is significant by ≤ 0.05 and ≤ 0.001 p-value, respectively.

Dataset	12			127			451		
	Min F1	Max F1	ES	Min F1	Max F1	ES	Min F1	Max F1	ES
Waseem	S_0: 0.676	S_{10}: 0.731	0.426*	S_5: 0.709	S_{15}: 0.726	0.131	S_0: 0.675	S_{10}: 0.723	0.390*
Davidson	S_{20}: 0.759	S_{15}: 0.791	0.441**	S_{10}: 0.755	S_{20}: 0.776	0.273*	S_0: 0.745	S_{15}: 0.786	0.491**
Founta	S_5: 0.872	S_{10}: 0.886	0.402*	S_5: 0.876	S_{20}: 0.888	0.356*	S_0: 0.874	S_0: 0.885	0.360*
OLID	S_{20}: 0.672	S_{10}: 0.718	0.207	S_0: 0.675	S_{15}: 0.725	0.169	S_0: 0.647	S_{10}: 0.731	0.287*
HateXplain	S_{20}: 0.634	S_{15}: 0.679	0.687**	S_5: 0.630	S_{20}: 0.674	0.637**	S_5: 0.636	S_{10}: 0.680	0.588**
Dynabench	S_5: 0.653	S_{20}: 0.660	0.153	S_5: 0.637	S_{15}: 0.659	0.468**	S_{15}: 0.623	S_{20}: 0.654 :	0.600**
Toxigen	S_{20}: 0.767	S_{10}: 0.771	0.180	S_5: 0.767	S_{10}: 0.771	0.218	S_5: 0.767	S_{10}: 0.771	0.228

to variation in seeding. *However, due to randomness in weight initialisation, the PLMs encode subjectivity across different datasets for hate detection.*

Table 6: A comparison of minimum and maximum macro F1 is obtained when the ps is constant but ms varies. ES stands for effect size. ** and * indicate whether the difference in minimum and maximum macro F1 is significant by ≤ 0.05 and ≤ 0.001 p-value, respectively.

Dataset	0			5			10			15			20		
	Min F1	Max F1	ES	Min F1	Max F1	ES	Min F1	Max F1	ES	Min F1	Max F1	ES	Min F1	Max F1	ES
Waseem	S_{451}: 0.675	S_{127}: 0.709	0.261	S_{12}: 0.691	S_{127}: 0.709	0.126	S_{127}: 0.714	S_{12}: 0.731	0.142	S_{12}: 0.711	S_{127}: 0.726	0.123	S_{12}: 0.686	S_{127}: 0.714	0.217
Davidson	S_{451}: 0.745	S_{127}: 0766.	0.232	S_{127}: 0.757	S_{12}: 0.763	0.090	S_{127}: 0.755	S_{12}: 0.772	0.221	S_{127}: 0.757	S_{12}: 0.791	0.435*	S_{451}: 0.755	S_{127}: 0.776	0.291*
Founta	S_{12}: 0.879	S_{451}: 0.885	0.204	S_{12}: 0.872	S_{127}: 0.876	0.123	S_{451}: 0.884	S_{127}: 0.887	0.093	S_{12}: 0.885	S_{127}: 0.887	0.087	S_{12}: 0.884	S_{127}: 0.888	0.121
OLID	S_{451}: 0.647	S_{127}: 0.675	0.089	S_{451}: 0.661	S_{12}: 0.689	0.106	S_{12}: 0.718	S_{451}: 0.731	0.056	S_{451}: 0.692	S_{127}: 0.725	0.141	S_{12}: 0.672	S_{451}: 0.703	0.113
HateXplain	S_{127}: 0.658	S_{12}: 0.674	0.215	S_{127}: 0.630	S_{12}: 0.6664	0.483**	S_{127}: 0.640	S_{451}: 0.680	0.504**	S_{127}: 0.660	S_{12}: 0.679	0.300*	S_{12}: 0.634	S_{127}: 0.674	0.591**
Dynabench	S_{451}: 0.648	S_{127}: 0.656	0.181	S_{127}: 0.637	S_{12}: 0.653	0.347*	S_{451}: 0.654	S_{127}: 0.657	0.06	S_{451}: 0.625	S_{127}: 0.659	0.701**	S_{127}: 0.634	S_{12}: 0.660	0.142
Toxigen	S_{12}: 0.769	S_{127}: 0.769	0.034	S_{451}: 0.767	S_{12}: 0.768	0.075	S_{12}: 0.771	S_{127}: 0.771	0.050	S_{127}: 0.770	S_{12}: 0.770	0.032	S_{12}: 0.767	S_{127}: 0.768	0.059

B.2 Seed-wise results for RQ3

Results controlling for ms are listed in Table 9.

B.3 Seed-wise results for RQ4

Tables 7, 8, and 10 enlist the per-seed comparison of performance for the layer and region finetuning, respectively. We observe that there is no lottery ticket to the best/most critical layer when examined from the point of view of ms, BERT-variants, and datasets.

Table 7: Comparison of seed S_{ms} and L_i^{th} layer combinations which lead to minimum and maximum macro-F1 for a dataset. ES stands for effect size. ** and * indicate whether the difference in minimum and maximum macro F1 is significant by ≤ 0.05 and ≤ 0.001 p-value, respectively.

Dataset	**BERT**			**BERTweet**			**HateBERT**			**mBERT**		
	Min F1	Max F1	ES	Min F1	Max F1	ES	Min F1	Max F1	ES	Min F1	Max F1	ES
Waseem	S_{12}, L_6: 0.758	S_{12}, L_{11}: 0.806	0.484**	S_{127}, L_6: 0.758	S_{127}, L_{11}: 0.810	0.944**	S_{451}, L_1: 0.752	S_{127}, L_{10}: 0.813	0.619**	S_{451}, L_9: 0.732	S_{127}, L_5: 0.793	0.617**
Davidson	S_{12}, L_{11}: 0.887	S_{451}, L_4: 0.931	0.854**	S_{12}, L_6: 0.899	S_{12}, L_5: 0.935	1.824**	S_{12}, L_{10}: 0.904**	S_{127}, L_5: 0.932	0.561**	S_{12}, L_{10}: 0.852	S_{451}, L_4: 0.922	1.367**
Founta	S_{12}, L_7: 0.916	S_{127}, L_5: 0.929	0.485**	S_{127}, L_0: 0.918	S_{451}, L_3: 0.930	0.486**	S_{12}, L_2: 0.915	S_{12}, L_9: 0.928	0.484**	S_{12}, L_{11}: 0.890	S_{12}, L_4: 0.924	1.120**
OLID	S_{127}, L_0: 0.732	S_{451}, L_{11}: 0.802	0.420*	S_{12}, L_0: 0.747	S_{127}, L_9: 0.817	0.438*	S_{451}, L_0: 0.738	S_{127}, L_8: 0.806	0.383*	S_{127}, L_{10}: 0.624	S_{451}, L_4: 0.764	0.595**
HateXplain	S_{451}, L_{11}: 0.639	S_{12}, L_10: 0.766	1.807**	S_{12}, L_6: 0.586	S_{12}, L_9: 0.770	2.616**	S_{12}, L_7: 0.638	S_{12}, L_4: 0.766	1.671**	$S_{451} L_9$: 0.615	S_{12}, L_7: 0.739	1.796**
Dynabench	S_{127}, L_6: 0.665	S_{451}, L_9: 0.756	2.082**	S_{12}, L_0: 0.705	S_{127}, L_{11}: 0.783	1.824**	S_{127}, L_0: 0.706	S_{451}, L_{11}: 0.770	1.564**	S_{12}, L_0: 0.635	S_{451}, L_4: 0.720	1.737**
Toxigen	S_{12}, L_0: 0.767	S_{12}, L_{11}: 0.806	2.126**	S_{12}, L_1: 0.0.786	S_{12}, L_{11}: 0.827	2.621**	S_{127}, L_0: 0.775	S_{127}, L_{11}: 0.816	2.386**	S_{451}, L_0: 0.746	S_{12}, L_4: 0.777	1.821**

Table 8: Comparison of minimum and maximum macro F1 obtained per ms per BERT-variant. ES stands for effect size. ** and * indicates whether the difference in minimum and maximum macro F1 is significant by ≤ 0.05 and ≤ 0.001 p-value, respectively.

Dataset	Seed	BERT			BERTweet			HateBERT			mBERT		
		Min F1	Max F1	ES	Min F1	Max F1	ES	Min F1	Max F1	ES	Min F1	Max F1	ES
waseem	12	L_6: 0.758	L_{11}: 0.806	0.484**	L_7: 0.723	L_{10}: 0.786	0.620**	L_0: 0.758	L_{10}: 0.813	0.558**	L_4: 0.736	L_{11}: 0.788	0.523**
	127	L_5: 0.760	L_4: 0.806	0.463	L_6: 0.700	L_{11}: 0.810	0.944**	L_1: 0.778	L_{10}: 0.813	0.392*	L_8: 0.744	L_5: 0.793	0.500**
	451	L_6: 0.760	L_4: 0.799	0.379*	L_1: 0.727	L_{11}: 0.788	0.528**	L_1: 0.752	L_{10}: 0.813	0.614**	L_9: 0.732	L_5: 0.790	0.582**
davidson	12	L_{11}: 0.887	L_1: 0.930	0.837**	L_6: 0.887	L_5: 0.936	0.895**	L_7: 0.908	L_3: 0.932	0.512**	L_{10}: 0.852	L_2: 0.920	1.36**
	127	L_2: 0.903	L_5: 0.928	0.480**	L_7: 0.900	L_3: 0.935	0.782**	L_{10}: 0.904	L_5: 0.932	0.561**	L_8: 0.888	L_5: 0.918	0.576**
	451	L_{10}: 0.889	L_4: 0.931	0.788**	L_7: 0.905	L_3: 0.935	0.671**	L_7: 0.906	L_4: 0.930	0.461**	L_{11}: 0.893	L_4: 0.923	0.618**
founta	12	L_7: 0.916	L_4: 0.929	0.488**	L_8: 0.921	L_4: 0.930	0.378*	L_2: 0.916	L_9: 0.928	0.484**	L_{11}: 0.890	L_4: 0.924	1.121**
	127	L_0: 0.920	L_5: 0.929	0.334*	L_0: 0.918	L_{11}: 0.928	0.401*	L_9: 0.923	L_4: 0.928	0.232	L_{10}: 0.908	L_5: 0.922	0.503**
	451	L_3: 0.921	L_4: 0.928	0.280*	L_6: 0.920	L_3: 0.930	0.441*	L_{11}: 0.916	L_2: 0.928	0.453	L_2: 0.904	L_4: 0.918	0.489**
olid	12	L_1: 0.742	L_9: 0.799	0.359*	L_0: 0.747	L_6: 0.805	0.388*	L_0: 0.744	L_7: 0.797	0.302*	L_8: 0.700	L_3: 0.750	0.220
	127	L_0: 0.732	L_8: 0.793	0.346*	L_0: 0.760	L_9: 0.817	0.323*	L_6: 0.750	L_8: 0.806	0.287*	L_{10}: 0.624	L_4: 0.755	0.509**
	451	L_2: 0.748	L_{11}: 0.802	0.321*	L_1: 0.764	L_5: 0.812	0.307*	L_0: 0.738	L_3: 0.804	0.388*	L_{10}: 0.681	L_4: 0.765	0.493**
hatexplain	12	L_4: 0.695	L_{10}: 0.766	1.054**	L_6: 0.586	L_9: 0.770	2.616**	L_7: 0.638	L_4: 0.766	0.1671**	L_{10}: 0.647	L_7: 0.739	0.1.33**
	127	L_9: 0.721	L_7: 0.763	0.580**	L_5: 0.717	L_9: 0.757	0.559**	L_4: 0.658	L_3: 0.763	1.470**	L_7: 0.616	L_5: 0.736	1.724**
	451	L_{11}: 0.639	L_4: 0.754	1.524**	L_2: 0.691	L_5: 0.761	1.024**	L_1: 0.723	L_{11}: 0.765	0.640**	L_9: 0.616	L_7: 0.737	1.782**
dynabench	12	L_0: 0.697	L_9: 0.746	1.108**	L_0: 0.705	L_9: 0.781	1.859**	L_1: 0.706	L_9: 0.765	1.414**	L_0: 0.635	L_4: 0.717	1.764**
	127	L_6: 0.665	L_{10}: 0.754	2.006**	L_0: .710	L_{11}: 0.783	1.614**	L_0: 0.706	L_{10}: 0.764	1.394**	L_7: 0.661	L_4: 0.719	1.316**
	451	L_2: 0.699	L_9: 0.756	1.335**	L_0: 0.711	L_9: 0.782	1.716**	L_0: 0.717	L_{11}: 0.770	1.257**	L_0: 0.691	L_4: 0.720	0.633**
toxigen	12	L_0: 0.767	L_{11}: 0.806	2.216**	L_1: 0.780	L_{11}: 0.812	2.026**	L_0: 0.780	L_{11}: 0.812	2.026**	L_0: 0.754	L_4: 0.777	1.34**
	127	L_0: 0.769	L_{11}: 0.803	2.044**	L_1: 0.788	L_{11}: 0.826	2.313**	L_0: 0.775	L_{11}: 0.816	2.396**	L_0: 0.746	L_5: 0.774	1.619**
	451	L_0: 0.768	L_{11}: 0.804	2.263**	L_1: 0.787	L_{11}: 0.826	2.551**	L_0: 0.778	L_{11}: 0.813	2.343**	L_0: 0.746	L_7: 0.775	1.619**

Table 9: Comparison of maximum macro F1 obtained under varying ms for the simple (S), medium (M) and complex (C) CH. $CH_{x,y}$ captures the difference in performance when comparing the given configuration under heads x and y. ES stands for effect size. ** and * indicates whether the difference in maximum macro F1 is significant by ≤ 0.05 and ≤ 0.001 p-value, respectively.

Dataset	BERT-variant	Seed	CH_S: F1	CH_M: F1	CH_C: F1	$CC_{S,M}$: ES	$CC_{M,C}$: ES	$CC_{C,S}$: ES
`Waseem`	BERT	12	0.703	0.752	0.773	0.481**	0.201	0.667**
		127	0.668	0.766	0.776	0.627**	0.066	0.704**
		451	0.697	0.765	0.767	0.533**	0.030	0.552**
	BERTweet	12	0.455	0.718	0.715	2.514**	0.016	2.463**
		127	0.454	0.734	0.731	2.939**	0.070	2.609**
		451	0.429	0.689	0.725	2.516**	0.343*	3.200**
	HateBERT	12	0.737	0.771	0.783	0.313*	0.119	0.433*
		127	0.751	0.781	0.787	0.236	0.073	0.319*
		451	0.752	0.775	0.779	0.254	0.019	0.280*
	mBERT	12	0.666	0.738	0.742	0.621**	0.014	0.622**
		127	0.639	0.742	0.750	0.896**	0.066	0.972**
		451	0.644	0.742	0.744	0.832**	0.026	0.858**
`Davidson`	BERT	12	0.781	0.722	0.811	0.800**	1.229**	0.453*
		127	0.768	0.789	0.811	0.272	0.290*	0.558**
		451	0.771	0.813	0.738	0.551**	0.905**	0.355*
	BERTweet	12	0.604	0.693	0.741	0.968**	0.480**	1.472**
		127	0.701	0.777	0.821	0.937**	0.602**	1.593**
		451	0.626	0.786	0.797	1.802**	0.165	1.979**
	HateBERT	12	0.824	0.842	0.850	0.275	0.148	0.423*
		127	0.825	0.832	0.818	0.111	0.186	0.070
		451	0.813	0.829	0.843	0.195	0.200	0.397*
	mBERT	12	0.724	0.759	0.723	0.428*	0.443*	0.018
		127	0.698	0.764	0.713	0.850**	0.670**	0.127
		451	0.713	0.723	0.754	0.135	0.389*	0.522**
`Founta`	BERT	12	0.891	0.892	0.892	0.030	0.010	0.040
		127	0.890	0.894	0.891	0.168	0.128	0.046
		451	0.892	0.893	0.894	0.028	0.042	0.069
	BERTweet	12	0.861	0.876	0.873	0.383*	0.080	0.301*
		127	0.855	0.879	0.873	0.693**	0.157	0.523**
		451	0.863	0.870	0.873	0.174	0.078	0.261
	HateBERT	12	0.886	0.888	0.890	0.047	0.074	0.126
		127	0.883	0.886	0.888	0.086	0.053	0.134
		451	0.881	0.884	0.885	0.074	0.040	0.118
	mBERT	12	0.840	0.849	0.846	0.224	0.058	0.162
		127	0.839	0.849	0.845	0.267	0.108	0.168
		451	0.840	0.852	0.848	0.327*	0.108	0.209
`OLID`	BERT	12	0.672	0.685	0.720	0.028	0.154	0.185
		127	0.675	0.708	0.672	0.165	0.185	0.023
		451	0.640	0.733	0.677	0.311*	0.149	0.145
	BERTweet	12	0.419	0.674	0.630	1.051**	0.160	0.817**
		127	0.506	0.722	0.608	1.015**	0.530**	0.412*
		451	0.453	0.707	0.582	0.966**	0.483**	0.455**
	HateBERT	12	0.659	0.742	0.730	0.421*	0.074	0.341*
		127	0.623	0.712	0.726	0.388*	0.097	0.503**
		451	0.674	0.699	0.726	0.147	0.113	0.260
	mBERT	12	0.507	0.555	0.591	0.172	0.162	0.328*
		127	0.538	0.617	0.647	0.239	0.117	0.348*
		451	0.574	0.614	0.504	0.125	0.353*	0.226
`HateXplain`	BERT	12	0.661	0.661	0.685	0.010	0.358*	0.363*
		127	0.677	0.679	0.676	0.045	0.037	0.009
		451	0.674	0.688	0.692	0.230	0.035	0.274
	BERTweet	12	0.621	0.663	0.655	0.551**	0.112	0.437*
		127	0.616	0.651	0.619	0.478**	0.430*	0.036
		451	0.626	0.680	0.683	0.764**	0.031	0.763**
	HateBERT	12	0.691	0.697	0.714	0.076	0.228	0.309*
		127	0.677	0.705	0.709	0.391*	0.067	0.450*
		451	0.708	0.715	0.724	0.097	0.150	0.238
	mBERT	12	0.655	0.660	0.663	0.052	0.047	0.101
		127	0.658	0.670	0.658	0.163	0.163	0.002
		451	0.647	0.654	0.637	0.086	0.240	0.155
`Dynabench`	BERT	12	0.658	0.673	0.663	0.316*	0.219	0.086
		127	0.648	0.637	0.681	0.226	0.851**	0.640**
		451	0.663	0.663	0.674	0.020	0.201	0.231
	BERTweet	12	0.622	0.628	0.564	0.128	1.271**	1.105**
		127	0.590	0.607	0.496	0.381*	2.464**	2.076**
		451	0.571	0.611	0.608	0.825**	0.065	0.771**
	HateBERT	12	0.686	0.707	0.703	0.493**	0.095	0.367*
		127	0.681	0.657	0.702	0.512**	0.969**	0.461*
		451	0.685	0.709	0.696	0.532**	0.282*	0.232
	mBERT	12	0.641	0.644	0.547	0.052	1.894**	1.908**
		127	0.577	0.648	0.649	1.621**	0.018	1.514**
		451	0.626	0.650	0.648	0.490**	0.036	0.459*
`Toxigen`	BERT	12	0.777	0.800	0.801	1.407**	0.052	1.468**
		127	0.776	0.802	0.802	1.450**	0.003	1.509**
		451	0.778	0.801	0.801	1.368**	0.000	1.407**
	BERTweet	12	0.753	0.770	0.770	0.898**	0.062	0.916**
		127	0.753	0.770	0.769	0.723**	0.027	0.670**
		451	0.753	0.771	0.772	1.033**	0.045	1.111**
	HateBERT	12	0.776	0.806	0.809	1.882**	0.182	1.986**
		127	0.777	0.807	0.808	1.557**	0.116	1.989**
		451	0.777	0.806	0.807	1.534**	0.070	1.539**
	mBERT	12	0.735	0.757	0.758	1.182**	0.061	1.233**
		127	0.736	0.757	0.758	1.228**	0.017	1.250**
		451	0.736	0.756	0.758	1.134**	0.140	1.329**

Table 10: Comparison of regional-wise macro F1 obtained under varying ms for the BERT-variants. We measure the impact on performance when a region R is set to trainable or unfrozen (T) vs. when it is non-trainable or frozen. ES stands for effect size. Further ** and * indicate whether the difference in macro F1 is significant by ≤ 0.05 and ≤ 0.001 p-value, respectively.

Dataset	BERT	SEED	R_1T	R_1F	R_1T/F:ES	R_2T	R_2F	R_2T/F:ES	R_3T	R_3F	R_3T/F:ES	R_4T	R_4F	R_4T/F:ES
Waseem	BERT	12	0.815	0.816	0.007	0.840	0.820	0.232	0.821	0.822	0.009	0.816	0.814	0.028
		127	0.795	0.822	0.298*	0.833	0.801	0.307*	0.786	0.811	0.245	0.803	0.831	0.297*
		451	0.831	0.812	0.189	0.824	0.822	0.015	0.828	0.811	0.186	0.824	0.813	0.078
	BERTweet	12	0.836	0.799	0.392*	0.814	0.827	0.130	0.831	0.820	0.086	0.812	0.823	0.085
		127	0.842	0.803	0.387*	0.831	0.812	0.279*	0.820	0.821	0.066	0.811	0.842	0.352*
		451	0.832	0.426	4.936**	0.844	0.819	0.283*	0.818	0.827	0.064	0.821	0.820	0.001
	HateBERT	12	0.799	0.812	0.083	0.831	0.823	0.107	0.817	0.812	0.086	0.818	0.799	0.207
		127	0.814	0.767	0.432*	0.809	0.820	0.114	0.815	0.829	0.129	0.818	0.828	0.146
		451	0.824	0.820	0.034	0.821	0.805	0.152	0.819	0.821	0.029	0.800	0.822	0.224
	mBERT	12	0.802	0.798	0.074	0.790	0.801	0.095	0.806	0.793	0.069	0.826	0.806	0.183
		127	0.799	0.805	0.037	0.763	0.802	0.370*	0.813	0.794	0.161	0.788	0.802	0.166
		451	0.791	0.786	0.022	0.812	0.738	0.733**	0.802	0.798	0.033	0.786	0.797	0.119
Davidson	BERT	12	0.926	0.921	0.116	0.919	0.924	0.096	0.454	0.930	13.303**	0.893	0.922	0.551**
		127	0.454	0.905	13.147**	0.454	0.921	13.839**	0.927	0.919	0.159	0.454	0.915	11.960**
		451	0.918	0.925	0.114	0.932	0.910	0.454*	0.454	0.932	14.392**	0.454	0.923	12.794**
	BERTweet	12	0.454	0.926	12.596**	0.454	0.935	13.664**	0.862	0.929	1.251**	0.454	0.931	14.453**
		127	0.454	0.924	12.368**	0.454	0.930	15.046**	0.454	0.933	15.144**	0.506	0.933	7.991**
		451	0.454	0.929	14.575**	0.454	0.934	15.645**	0.454	0.926	13.377**	0.454	0.882	9.952**
	HateBERT	12	0.454	0.919	12.211**	0.454	0.919	12.672**	0.454	0.920	13.229**	0.454	0.928	13.370**
		127	0.924	0.924	0.037	0.454	0.934	13.568**	0.454	0.911	12.876**	0.454	0.922	12.962**
		451	0.454	0.454	0.000	0.917	0.917	0.026	0.454	0.920	12.774**	0.454	0.919	13.289**
	mBERT	12	0.454	0.913	12.393**	0.454	0.925	12.538**	0.483	0.923	9.358**	0.454	0.923	13.992**
		127	0.454	0.902	12.214**	0.454	0.916	13.964**	0.454	0.913	10.779**	0.454	0.923	13.322**
		451	0.454	0.921	12.280**	0.476	0.916	9.423**	0.457	0.924	11.758**	0.454	0.920	13.139**
Founta	BERT	12	0.435	0.875	16.947**	0.435	0.903	22.165**	0.435	0.435	0.000	0.435	0.906	20.983**
		127	0.435	0.435	0.000	0.435	0.435	0.000	0.435	0.435	0.000	0.904	0.435	21.930**
		451	0.435	0.901	20.681**	0.435	0.904	21.262**	0.435	0.435	0.000	0.435	0.435	0.000
	BERTweet	12	0.435	0.435	0.000	0.435	0.435	0.000	0.435	0.435	0.000	0.435	0.755	9.979**
		127	0.435	0.435	0.000	0.435	0.435	0.000	0.435	0.435	0.000	0.435	0.435	0.000
		451	0.435	0.435	0.000	0.435	0.435	0.000	0.435	0.435	0.000	0.435	0.553	3.100**
	HateBERT	12	0.435	0.435	0.000	0.435	0.435	0.000	0.435	0.910	19.936**	0.435	0.435	0.000
		127	0.435	0.435	0.000	0.435	0.915	24.121**	0.435	0.873	16.404**	0.435	0.871	15.600**
		451	0.435	0.435	0.000	0.435	0.905	22.709**	0.435	0.794	11.383**	0.435	0.889	19.753**
	mBERT	12	0.435	0.834	13.390**	0.758	0.435	9.584**	0.435	0.877	16.618**	0.435	0.435	0.000
		127	0.435	0.435	0.000	0.435	0.854	14.137**	0.435	0.435	0.000	0.435	0.435	0.000
		451	0.435	0.895	19.070**	0.435	0.435	0.000	0.435	0.435	0.000	0.435	0.909	20.701**
OLID	BERT	12	0.737	0.740	0.008	0.773	0.795	0.075	0.777	0.767	0.008	0.778	0.790	0.081
		127	0.755	0.762	0.052	0.786	0.765	0.103	0.783	0.775	0.093	0.767	0.785	0.085
		451	0.771	0.777	0.019	0.768	0.800	0.186	0.771	0.798	0.061	0.775	0.794	0.134
	BERTweet	12	0.774	0.419	1.535**	0.808	0.803	0.020	0.419	0.825	1.950**	0.773	0.815	0.307*
		127	0.792	0.419	1.644**	0.419	0.814	1.776**	0.419	0.815	1.846**	0.419	0.812	1.797**
		451	0.804	0.419	1.704**	0.810	0.811	0.048	0.790	0.806	0.155	0.419	0.804	1.749**
	HateBERT	12	0.787	0.479	1.254**	0.419	0.770	1.409**	0.764	0.765	0.015	0.770	0.795	0.187
		127	0.749	0.749	0.042	0.776	0.788	0.047	0.756	0.762	0.050	0.751	0.789	0.239
		451	0.769	0.766	0.023	0.795	0.793	0.024	0.783	0.787	0.062	0.419	0.765	1.435**
	mBERT	12	0.715	0.735	0.094	0.681	0.678	0.057	0.704	0.775	0.244	0.740	0.769	0.163
		127	0.780	0.727	0.230	0.707	0.763	0.266	0.419	0.756	1.276**	0.758	0.761	0.015
		451	0.419	0.419	0.000	0.764	0.771	0.035	0.432	0.772	1.343**	0.730	0.736	0.069
HateXplain	BERT	12	0.747	0.746	0.004	0.769	0.776	0.133	0.431	0.753	4.846**	0.762	0.758	0.058
		127	0.770	0.393	7.340**	0.718	0.783	0.945**	0.393	0.721	5.729**	0.733	0.750	0.240
		451	0.769	0.758	0.180	0.747	0.776	0.400*	0.393	0.779	8.101**	0.759	0.702	0.817**
	BERTweet	12	0.775	0.393	7.975**	0.767	0.775	0.137	0.393	0.787	8.366**	0.393	0.769	6.704**
		127	0.739	0.393	6.839**	0.393	0.501	2.289**	0.393	0.779	8.771**	0.393	0.794	8.514**
		451	0.394	0.393	0.052	0.739	0.778	0.549**	0.393	0.791	8.459**	0.393	0.722	5.741**
	HateBERT	12	0.758	0.752	0.099	0.755	0.780	0.406*	0.753	0.771	0.271	0.739	0.751	0.159
		127	0.768	0.754	0.144	0.725	0.757	0.411*	0.762	0.770	0.150	0.761	0.760	0.012
		451	0.760	0.393	6.310**	0.747	0.776	0.407*	0.768	0.777	0.129	0.737	0.780	0.695**
	mBERT	12	0.739	0.719	0.285*	0.393	0.732	7.035**	0.582	0.736	2.129**	0.676	0.721	0.676**
		127	0.740	0.393	6.895**	0.593	0.639	0.598**	0.682	0.752	0.934**	0.393	0.738	6.722**
		451	0.734	0.745	0.179	0.732	0.737	0.090	0.393	0.746	7.218**	0.719	0.731	0.198
Dynabench	BERT	12	0.317	0.349	1.573**	0.349	0.318	1.506**	0.349	0.768	12.256**	0.349	0.760	12.166**
		127	0.349	0.349	0.000	0.349	0.732	11.640**	0.349	0.713	12.153**	0.317	0.771	13.692**
		451	0.349	0.349	0.000	0.349	0.688	10.104**	0.349	0.349	0.000	0.349	0.771	13.173**
	BERTweet	12	0.498	0.349	3.944**	0.349	0.349	0.000	0.349	0.765	14.378**	0.349	0.795	15.670**
		127	0.317	0.317	0.000	0.349	0.730	10.885**	0.349	0.349	0.000	0.349	0.813	15.126**
		451	0.349	0.349	0.000	0.317	0.349	1.571**	0.349	0.777	14.698**	0.349	0.392	1.469**
	HateBERT	12	0.349	0.691	10.451**	0.349	0.349	0.000	0.349	0.775	13.318**	0.349	0.781	14.576**
		127	0.349	0.349	0.000	0.349	0.727	11.989**	0.349	0.752	11.896**	0.349	0.785	13.631**
		451	0.317	0.349	1.571**	0.349	0.748	12.493**	0.349	0.742	10.092**	0.349	0.787	13.536**
	mBERT	12	0.349	0.367	0.673**	0.349	0.349	0.009	0.349	0.666	9.274**	0.349	0.716	10.999**
		127	0.349	0.619	7.138**	0.349	0.349	0.000	0.349	0.675	9.671**	0.349	0.723	12.271**
		451	0.317	0.349	1.571**	0.349	0.380	1.141**	0.349	0.709	9.804**	0.349	0.724	11.119**
Toxigen	BERT	12	0.333	0.333	0.045	0.333	0.333	0.000	0.333	0.333	0.000	0.333	0.333	0.045
		127	0.333	0.333	0.000	0.333	0.333	0.045	0.333	0.333	0.000	0.333	0.333	0.000
		451	0.333	0.333	0.000	0.333	0.333	0.000	0.333	0.333	0.045	0.333	0.333	0.045
	BERTweet	12	0.333	0.333	0.045	0.333	0.333	0.000	0.333	0.333	0.045	0.333	0.333	0.000
		127	0.333	0.333	0.000	0.333	0.333	0.045	0.333	0.333	0.000	0.333	0.333	0.045
		451	0.333	0.333	0.045	0.333	0.333	0.045	0.333	0.333	0.000	0.333	0.333	0.045
	HateBERT	12	0.333	0.333	0.000	0.333	0.333	0.045	0.333	0.599	16.715**	0.333	0.333	0.000
		127	0.333	0.333	0.045	0.333	0.333	0.000	0.333	0.333	0.045	0.333	0.333	0.045
		451	0.333	0.333	0.000	0.333	0.333	0.000	0.333	0.333	0.045	0.333	0.333	0.045
	mBERT	12	0.333	0.333	0.000	0.333	0.333	0.045	0.333	0.333	0.000	0.333	0.333	0.000
		127	0.333	0.333	0.000	0.333	0.333	0.000	0.333	0.333	0.000	0.333	0.333	0.045
		451	0.333	0.333	0.000	0.333	0.333	0.000	0.333	0.333	0.045	0.333	0.333	0.000

C A case study of using DiVA

In this section, we consider a standard dataset, called LastFM-Asia, of social network users, first introduced in the FEATHER network analysis (Rozemberczki and Sarkar, 2020). We run some diffusion simulations on this.

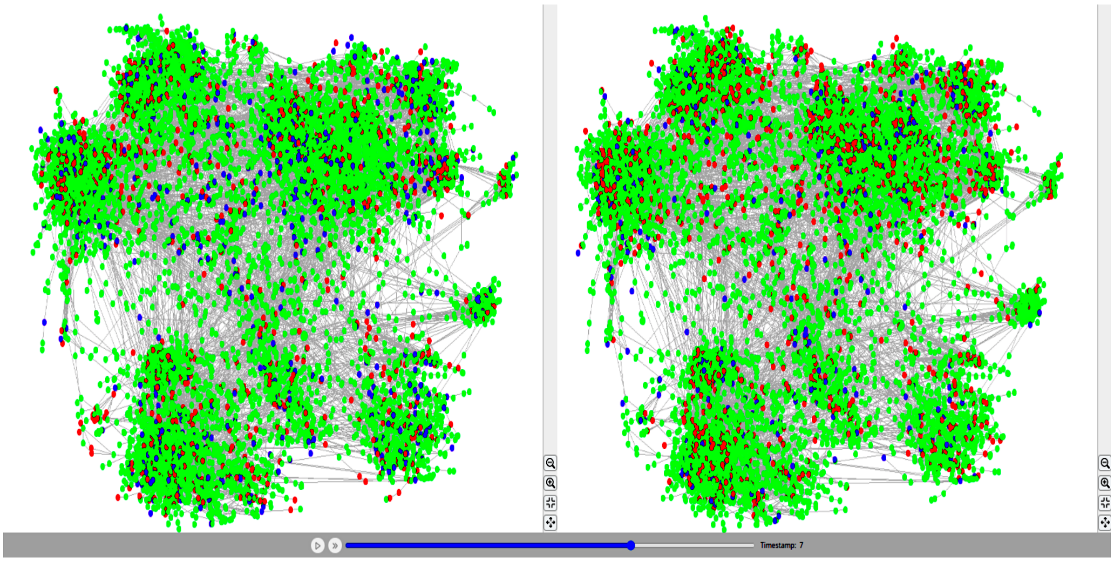


Figure 1: Results of our simulations on the FEATHER network. These observations are for $t = 7$ in the `Split View` of the dual visualisation mode. The green, red and blue nodes represent the susceptible, infected and recovered nodes, respectively. The recovery rate for the SIR model on the left is $\gamma = 0.1$. Meanwhile, keeping other factors constant, the recovery rate for the SIR model on the right is $\gamma = 0.05$.

Dataset. The dataset was curated from LastFM public API (March 2020) for the Asian region, consisting of users with 18 country labels. There are 7624 user nodes, which are connected via 27806 mutual friendship links.

Aim. Assume a case of an Internet virus that is spreading on the LastFM-Asia network. How many systems can we recover with varying rates of recovery?

Diffusion algorithm. In order to obtain recovered nodes in the network, the closest diffusion model would be the SIR (susceptible-infected-recovered) model. The SIR model has two parameters – (i) the rate of infection (β), which in our case is the rate at which the computer virus is affecting the users, and (ii) the rate of recovery (γ), which in our case is the rate at which users can recover their systems.

Simulation environment. All the simulations are run on a locally hosted Chrome browser (Version 101.0.491.64) on a ThinkPad E480 with 16GB RAM.

Running simulations

- Bring up an instance of `DiVA`, and load the LastFM-Asia network.
- Switch to `Compare View` tab.
- Select SIR as the first diffusion algorithm with parameters $\beta_1 = 0.05$, $\gamma_1 = 0.1$.
- Select SIR as the first diffusion algorithm with parameters $\beta_2 = 0.05$, $\gamma_2 = 0.05$.

- For our use simulation we set the `Maximum Iterations` $= 10$, `Fraction Infected` $= 0.1$.
- Run the iteration and visualise the results with different γ rates.

Evaluating simulations

- Firstly, we obtain the network statistics as reported in the original paper, and as evident from Table 11(a), we can reproduce the same.
- Secondly, from Table 11(b), we notice that all things being equal, a higher rate of recovery translates to more nodes saved and restored from the computer virus. At the end of the simulation, we will be able to recover almost $1.5x$ the number of nodes (806 vs. 457) for a higher recovery rate. The same can be observed from the difference in the number of infected vs. recovered nodes from the infection plots in Figure 2. At timestamp $t = 2$, the delta increase in the number of infected nodes is 42, while at the end of the simulation ($t = 10$), the delta increase in the number of infected nodes is as high as 536.
- For a SIR model with green, blue, and red nodes specifying the susceptible, infected, and recovered nodes, respectively, we can visually observe the impact of the high recovery rate in the first SIR model by the higher number of blue nodes in the left split of Figure 1. Complementary information for the lower rate of recovery in the second SIR model can be visually observed by a lower number of blue nodes in the left split of Figure 1. The right split subsequently has a higher volume of red nodes.

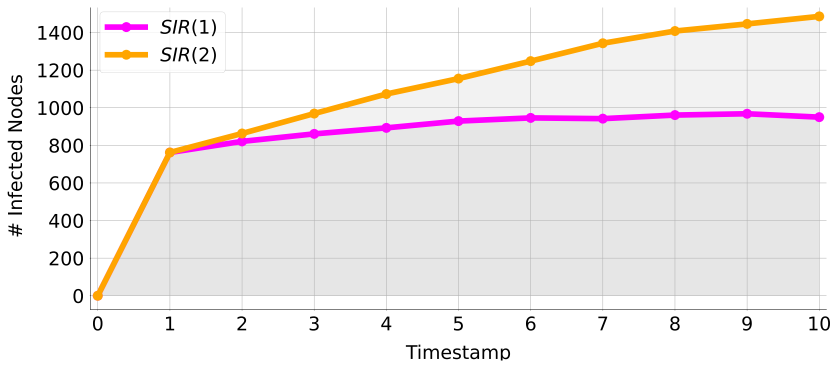


Figure 2: The number of infected nodes for the two simulations as obtained from the `Report View Tab`. SIR (1) and SIR (2) refer to models with recovery rates 0.1 and 0.05, respectively.

Table 11: Evaluating the results for simulation: (a) Reporting the reproduced network statistics. (b) Listing out the number of recovered nodes in the system for different rates of recoveries (γ).

(a)

Statistics	**Original**	**Reproduced**
# Nodes	7624	7624
# Edges	27806	27806
Density	0.0009	0.0009
Transitivity	0.179	0.179

(b)

Timestamp	**# Recovered Nodes**	
	$\gamma = 0.1$	$\gamma = 0.05$
0	0	0
1	0	0
2	73	21
3	153	57
4	238	95
5	326	154
6	426	207
7	533	260
8	624	324
9	710	389
10	806	457

D Some famous cases of name-calling

Political leaders are often given nicknames depending on whether they receive the name from their supporters or the opposition. For example, Prime Minister Narendra Modi is affectionately referred to as “Sher” *(Lion)* by his supporters; meanwhile, INC leader Rahul Gandhi is frequently called “Pappu” *(dim-witted)* by the opposition. Back in the year 2017, when opposition leader Rahul Gandhi remarked, “They say, you give me potatoes, and I will pass them through a machine turning them into gold,” he was rhetorically referring to Prime Minister Narendra Modi. However, a cropped version of Rahul’s speech, where only the words, “You give me potatoes, and I will pass them through a machine turning them into gold,” was widely circulated on the web. A political gibe backfired for Rahul Gandhi, who, to date, is trolled for this clip. Such is the nature of Indian politics! Similarly, during the Assembly elections, we notice some new and old nicknames surface on Twitter.

Yogi and “bulldozer”. BJP leader and UP’s CM Yogi Adityanath is often referred to as “Baba” *(sage)*, given that he was a saint before joining politics. In our study, we observe the term “baba” being used as a source of affection as well as irony. Consequently, “Bulldozer Baba” *(the bulldozer saint)* refers to his act of bulldozing the properties of apparent criminals and unlawful citizens. While the opposition asks if bulldozers and the destruction of the property could improve the financial conditions in UP, his supporters are presenting him with gold-plated bulldozers to cement his rule with an iron fist. Upon Adityanath’s re-election as the CM of UP in 2022, his supporters flooded social media with images and videos of bulldozers. Some are even seen standing atop bulldozers and chanting victory to symbolise that nothing could come in the way of the bulldozer, like the determination of Yogi Adityanath. The tweet, “बुलडोजर बाबा से मन का सवाल। क्या बुलडोजर से बेरोजगारी दूर हो जायेगी? किसान को फसल का दाम मिल जायेगा? महगाई खत्म हो जायेगी?” *(I have the following question in my mind for Bulldozer Baba, will bulldozers solve unemployment, give farmers their due and reduce inflation?)*, captures the direct attack employing the nickname “Bulldozer Baba”. On the other hand, we also see examples where tweets put a positive spin on the nickname (a) “बुलडोजर का बढता क्रेज। गोरखपुर के प्रमुख व्यापारी ने उत्तर प्रदेश मुख्यमंत्री योगी आदित्यनाथ जी को चांदी का बुलडोजर भेंट किया।” *(Bulldozer is gaining popularity. A prominent businessman from Gorakhpur gifts Yogi Adityanath a silver-plated bulldozer)*, and (b) “यूपी में सरकार बनाने जा रही बीजेपी, हेमा मालिनी बोलीं- बुलडोजर के आगे कुछ नहीं आ सकता” *(BJP will be forming the government in UP, Hema Malini says - Nothing can stand in the bulldozer’s way.)*

Channi “sand mafia”. In Punjab, INC, BJP, and AAP indulge in hefty name-calling. Opposition extensively uses the term “RetaChorChanni” *(Sand Mafia Channi)* to refer to Charanjit Singh Channi, Punjab’s then CM, for his apparent implication in the case of sand mining scams and corruption. The damaging #RetaChorChanni campaign against Channi did prove advantageous. What started as a slanderous remark got bolstered when reports of corruption probes against him and his associates came to light. Some posts attacking his leadership are: (a) “जिस चन्नी को राहुल गांधी कोहिनूर बताते है, पंजाब की जनता उसे रेता चोर बुलाती है। सोचने वाली बात है। सिर्फ 111 दिन के राज में घर से 10 Crore Cash बरामद हुए #RetaChorChanni” *(The very Channi whom Rahul Gandhi calls a gem, the people of Punjab call him a Sand Mafia. In his 111 days of ruling, it is pretty astonishing how he amassed 10 Crore Cash.)* and (b) “CONgRSS promotes corrupt people. Their top leadership facilitates corrupt and incompetent people. Siddhu compromised on a Sand Mafia being a CM face from your Congress. #RetaChorChanni #RetaChorMachayeShor" *(Sand Mafia crying wolf)*. The opposition's constant remarks on #RetaChorChanni even led news media to question his importance in the INC alliance with headlines like "पंजाब में कांग्रेस के लिए चन्नी जरूरी या मजबूरी?" *(Is Chennai an asset or a burden for Congress in Punjab?)*.

Kejriwal "terrorist". Opposition parties call AAP's leader, Arvind Kejriwal, a "terrorist." Puns suggest that the Aam Aadmi Party (the party of/by commoner) is rather "अरविंद आतंकवादी पार्टी" *(Arvind Terrorist Party)*, and this is widely circulated. The tweets can be seen as equating Kejriwal and his party to a terrorist organisation. He has turned this smear campaign to his advantage by saying he is the world's first "sweet terrorist" who works for his people's welfare. He has put a positive spin on it by trying to showcase his party's work in Delhi and what they hope to replicate and improve in Punjab and Goa. With banters such as "जिसको पप्पू और गप्पू आतंकवादी कह रहे है, वही केजरीवाल जी देश के बच्चो के लिए हजाअरो स्मार्ट क्लासरूम बना रहे है। फर्क साफ है जो भी अच्छी शिक्षा फ्री बिजली अच्छे हस्पिटल की बात करेगा उससे भाजपा कांग्रेस मिलकर लडएंगी।" *(The pappus and gappus are calling me a terrorist. I am that terrorist who has installed smart classrooms for the kids. The difference is clear: anyone who speaks of quality education, free electricity, and good hospitals, BJP+INC, will unite to fight that person.)* Kejriwal can be seen mockingly bashing the opposition for trying to sully any leader like him who speaks of the welfare of the people.

The symbolic "lal topi". SP leader Akhilesh Yadav always dons a "Lal Topi" *(red cap)* to symbolise his party's socialist ideology and the struggles of the downtrodden. Conversely, the opposition implicitly uses "Lal Topi" to symbolise corruption, red-tapism, and violence. For example, BJP leaders utilise "Laal Topi" to be a connotation of Akhilesh's criminal affiliations and occasionally make remarks like –"लाल टोपी का मतलब दंगा, लाल टोपी का मतलब हिस्ट्रीशीटर, लाल टोपी का मतलब राह चलते नागरिकों के साथ रहजनी, लाल टोपी का मतलब किसानों के खेत से ट्यूबेल और पंपसेट का चोरी हो जाना" *(The red cap represents riots, the red cap represents hardened criminals, the red cap represents looting and snatching on streets, and the red cap represents stealing of hand pumps and tube wells from farm fields)*.

E Generalisability of `FiADD`

We also explore the generalisation capabilities of the proposed framework on other semantically similar tasks in NLP where the surface and implied forms differ. To this end, we explore three standard SemEval tasks for our generalisability analysis – sarcasm, stance and irony detection.

Datasets. Sarcasm detection (Abu Farha *et al.*, 2022) and irony detection (Van Hee *et al.*, 2018) are two-way classification datasets. Meanwhile, stance detection (Mohammad *et al.*, 2016) is a three-way classification. While we have implied annotations for sarcasm, they are missing for the other two datasets. Here, no additional annotations are performed. An overview of the three datasets is available in Table 12.

Table 12: The class-wise distributions for (i) Sarcasm: non-sarcastic (N-SAR) or sarcastic (SAR), (ii) Irony: unironic (N-IRO) or ironic (IRO), (iii) Stance: neutral (NEU) or against (ANG) or in-favour (FAV).

Task	**Labels**		
	N-SAR	SAR	
Sarcasm	3801	1067	
	N-IRO	IRO	
Irony	1890	1756	
	NEU	ANG	FAV
Stance	918	1969	982

Experimental setup. For the SemEval tasks, we consider BERT and XLM (Chi *et al.*, 2022) for evaluation based on their popularity in the SemEval. The PLM variants are 'bert-base-uncased' for BERT and 'xlm-roberta-large' for XLM. The availability of implied annotations in the sarcasm dataset enables us to access `FiADD`'s $ADD^{Inf+foc}$ variant. In the other two tasks, we experiment with only the ADD^{foc} variant.

Observations. We observe reasonable improvements in macro-F1 (0.4%-2.4%) across all three tasks using both PLMs (Table 13(a-b)). Further, for the minority class considering the best of the BERT and XLM, we observe `FiADD`variants report an improvement of 6.06% ($\uparrow$ 23.96)%, 1.35% ($\uparrow$ 2.65%) and 3.14% ($\uparrow$ 5.42%) respectively in sarcasm, stance and irony detection.

Table 13: Performance comparison of ACE vs `FiADD`'s variants on sarcasm, irony and stance detection. We report each model's best performance based on overall macro-F1 out of three random seeds.

(a)

Tasks	Model	Metric (F1)	Objective: ACE	Objective: ACE+ADD$^{\text{inf + foc}}$
Sarcasm	BERT	Macro	**0.5651**	0.5610
		N-SAR	0.8329	0.8501
		SAR	0.2974	0.2720
	XLM	Macro	0.5577	**0.5814**
		N-SAR	0.8626	0.8493
		SAR	0.2529	0.3135

(b)

Tasks	Model	Metric (F1)	Objective: ACE	Objective: ACE+ADD^{foc}
Irony	BERT	Macro	0.6886	**0.7099**
		N-IRO	0.7379	0.7658
		IRO	0.6394	0.6540
	XLM	Macro	0.6637	**0.6740**
		N-IRO	0.7482	0.7372
		IRO	0.5792	0.6106
Stance	BERT	Macro	0.5735	**0.5776**
		NEU	0.5009	0.4910
		ANG	0.6964	0.7090
	XLM	Macro	0.5560	**0.5692**
		NEU	0.4494	0.4637
		ANG	0.7084	0.7199
		FAV	0.5103	0.5238

F Case study on knowledge-drift

Lexical debiasing. To overcome lexical bias in hate speech detection, (Badjatiya *et al.*, 2019) employed a knowledge-based generalisation method. It involved the replacement of the bias-sensitive words (BSW) in the training dataset with an ancestor from the WordNet (Miller, 1995) hypernym-tree.

Knowledge-drift hypothesis. Motivated by the energy-bias analogy (Section 4.4.4), we conjecture that replacing all occurrences $w \in BSW$ with its wordnet ancestor $a \in A$ will shift the bias from w to a. While generalisation can be true for all the debiasing methods we discuss in this survey, for this experiment, we focus on the WordNet-based substitution method for lexical debiasing.

Limited evaluation. Post substitution of W with A, the authors employed the BSW on the original W to evaluate bias. There is no discussion on evaluating the bias in terms of A, which is now a substitute for W.

Experimental setup. All experiments are carried out on a Ubuntu 18.04.5 LTS system with $126G$ RAM and $32G$ Tesla V100. We apply the debiasing method on the W&H dataset (Waseem and Hovy, 2016) to verify the hypothesis and compare by training a BERTweet (Nguyen *et al.*, 2020) classifier.

Table 14: (a) BSWs w with their corresponding selected generalisations a according to the Wordnet-3 scheme (Badjatiya *et al.*, 2019). (b) pB values for the set of BSWs W and A for (i) the model obtained on the original Waseem dataset M_{bias} and (ii) the model obtained after lexical database generalisation M_{gen}.

(a)

$w \in W$	$a \in A$
Muslim, prophet, woman, Christian, terrorist, slave, man, child, driver, being	someone
Feminist, civilian, liar, comedian, god	
Hate, slavery, hatred, want, truth, freedom, and state	

(b)

Metric	M_{bias}	M_{gen}
pB_W	0.027	0
pB_A	0.002	0.016

Observation. Table 14(a) lists the replaced words and their respective generalisations. The observations in Table 14(b) indicate a shift of bias from source to target words. Since the terms in the set A are more likely to be present in non-toxic comments, this shift in bias can also be detrimental to the non-toxic class. While this is a proof of concept, it can be extrapolated that other mitigation techniques based on knowledge-based generalisations suffer from a similar shift of bias (Badjatiya *et al.*, 2019) instead of actually debiasing the dataset.

Resolving bias shift. Note that this shift of lexical bias is towards a more general set of terms A, where debiasing on this generalised dataset through upsampling (Dixon *et al.*, 2018) directly from an OSN (such as Twitter) seems like a feasible next step. For example, newly scraped comments containing terms such as *muslim* can not be directly added to the training dataset. However, if *muslim* is first generalised to *being*, randomly scraping new comments with this new keyword can lead to a significantly lower toxicity ratio.

G Additional configurations for in-domain attributes

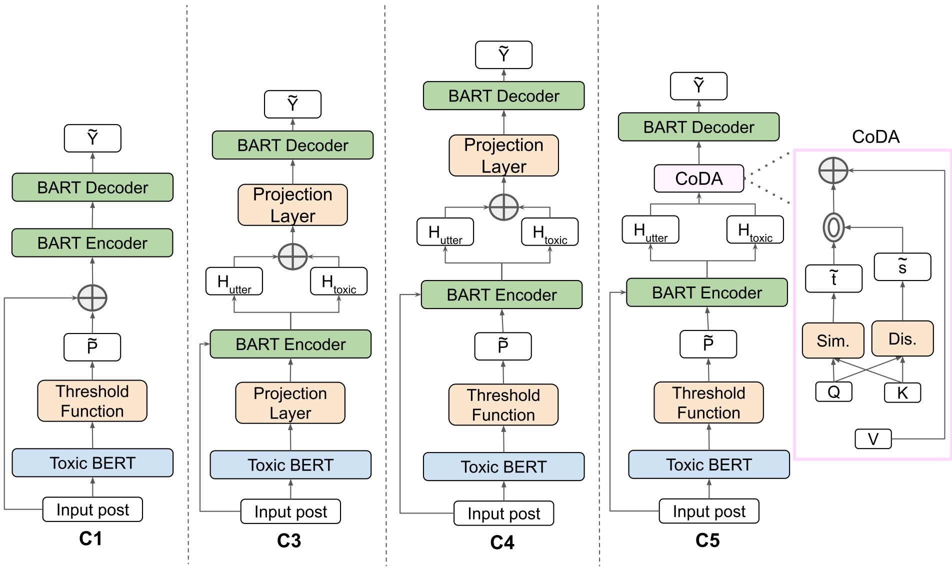


Figure 3: Configurations (C1, C3-C5) for incorporating *in-domain* attributes via the `ToxicBERT` regressor. BART encoded toxic attributes, and input representations are $\mathrm{H_{toxic}}$ and $\mathrm{H_{utter}}$, respectively. $\tilde{\mathcal{P}}$ is the modified toxic attributes vector, whereas $\tilde{Y}$ is the system generated explanations. The toxic-attributed BART so finetuned is our proposed system - `Tox-BART`. In C5, Sim. (Dis.) captures query (Q), Key (k), and value (V) for the similarity (dissimilarity) matrices. $\tilde{t}$ and $\tilde{s}$ represent the tanh and sigmoid functions, respectively. $\bigoplus$ and $\odot$ represent the concatenation/addition and multiplication operations, respectively.

We discuss three additional configurations that we study for infusing the *in-domain* attributes.

Configuration 3 (C3). We began with the very rudimentary concatenation of $\mathcal{P}$ with input $\mathcal{X}$. We first transform $\mathcal{P}$ into a higher-dimensional vector $\tilde{\mathcal{P}}$. This vector and the incoming posts are separately passed through the BART encoder, and the resultant latent embedding (H_{toxic} and H_{utter}) are concatenated and passed through another linear transformation to downsize before feeding to the decoder. The set of Equations 2 outlines the setup where $\mathcal{V}(.)$ refers to a linear transformation, and $[,]$ corresponds to the concatenation operation.

$$
\begin{gathered}
H_{toxic} = \mathcal{F}_{\theta}(\mathcal{V}_{6\times d}(\mathcal{P})); \quad H_{utter} = \mathcal{F}_{\theta}(\mathcal{X}) \\
\tilde{\mathcal{Y}} = \mathcal{G}_{\theta}(\mathcal{V}_{2d\times d}([H_{toxic}, H_{utter}]))
\end{gathered} \tag{2}
$$

Here, H_{utter} and H_{toxic} are the encoded representations of the input and the corresponding probability-to-special text tokens.

Configuration 4 (C4). We first to encode $\tilde{\mathcal{P}}$ and then concatenate. This will require the concatenated vector to undergo a linear transformation to match the decoder dimension. The set of Equations 3 outlines this setup.

$$
\begin{gathered}
H_{toxic} = \mathcal{F}_{\theta}(\Gamma(\lambda, \mathcal{P})); \quad H_{utter} = \mathcal{F}_{\theta}(\mathcal{X}) \\
\tilde{\mathcal{Y}} = \mathcal{G}_{\theta}(\mathcal{V}_{2d\times d}([H_{toxic}, H_{utter}]))
\end{gathered} \tag{3}
$$

Configuration 5 (C5). Building upon the previous configuration, here, instead of directly concatenating the two encoder outputs, we use the Compositional De-Attention framework (CoDA) (Tay *et al.*, 2019). CoDA determines the attention scores between the two encoder outputs. The intuition for this method is that some toxic attributes might be more critical or "similar" for some token in the utterance than others, which can be considered "dissimilar." The CoDA attention outputs are then combined (via addition) with the encoder outputs of input utterances before passing through the decoder. Equations 4 outline the setup.

$$\begin{aligned} H_{toxic} = \mathcal{F}_{\theta}(\Gamma(\lambda, \mathcal{P})); \quad H_{utter} = \mathcal{F}_{\theta}(\mathcal{X}) \\ \tilde{H} = H_{utter} + \psi(H_{toxic}, H_{utter}); \quad \tilde{\mathcal{Y}} = \mathcal{G}_{\theta}(\tilde{H}) \end{aligned} \tag{4}$$

where ψ refers to the CoDA framework (Tay *et al.*, 2019) that captures the attention score via $\psi = (tanh(\frac{QK^{\top}}{\sqrt{d_k}}) \odot sigmoid(\frac{\Phi(Q,K)}{\sqrt{d_k}}))V$.

Performance on additional configurations

Table 15 shows that $\texttt{Tox-BART}_{\text{C3}}$ performs worse than even vanilla BART and GPT-2. We conjecture this arises from the difference in the distribution space of probability scores vectors and BART representations. On the other end of the spectrum, we observe for $\texttt{Tox-BART}_{\text{C5}}$ that attentive concatenation may be overfitting the toxicity signals, leading to a loss of information. The lower efficacy of $\texttt{Tox-BART}_{\text{C5}}$ aligns with previous research on attention-based KG-tuple concatenation (Sridhar and Yang, 2022). Nevertheless, concatenation in the embedding space post encoding is not as effective as concatenation in the input space as in $\texttt{Tox-BART}_{\text{C1}}$.

Table 15: $\texttt{Tox-BART}_{\text{C1}}$ on `SBIC` for additional in-domain configurations.

Method	**BLEU**	**ROUGE-L**	**BERTScore**
$\texttt{Tox-BART}_{\text{C1}}$	64.89	63.83	64.52
$\texttt{Tox-BART}_{\text{C3}}$	12.89	17.39	34.06
$\texttt{Tox-BART}_{\text{C4}}$	0.76	4.77	34.94
$\texttt{Tox-BART}_{\text{C5}}$	61.77	65.71	82.51

H Algorithm for NACL

Algorithm 3 shows the learning protocol of NACL.

Algorithm 3 Learning NACL

Input: A set of hateful samples $T = \{t_1, ..., t_m\}; \forall j \quad \phi_{t_j} \geq \tau$
Output: A set of normalised samples $T' = \{t'_1, ..., t'_m\}; \forall j \quad \phi_{t'_j} < \tau$

Require: NACL(T, τ)
Pretrain HSI(T) and HIP(T)
repeat
for all $t \in T$ **do**
$[\langle s_1, e_1\rangle, .., \langle s_k, e_k\rangle] \leftarrow$ HSI(t) ▷ *Get indices of hate spans.*
$t' \leftarrow$ HIR($t, \langle s_1, e_1\rangle, .., \langle s_k, e_k\rangle$) ▷ *Generator: Get normalised text.*
$\phi_{t'} \leftarrow$ HIP(t') ▷ *Discriminator: Get the intensity of the normalised text.*
$R_{t'} = \tau - \phi_{t'}$ ▷ *Calculate reward for the generator - +ve, if intensity less than threshold, -ve, otherwise.*
$\mathcal{L} = \ell + (1 - R)$ ▷ *ℓ = Generator loss*
HIR $\leftarrow$ Backpropagation($\mathcal{L}$)
end for
until <termination-condition> **return** T'

Require: HIP(t)
$H' \leftarrow$ BiLSTM(BiLSTM(t))
$\alpha \leftarrow$ Attention(H')
$H = H' \cdot \alpha$
$\phi_t \leftarrow$ Linear(H) **return** ϕ_t ▷ *Hate intensity of the tweet.*

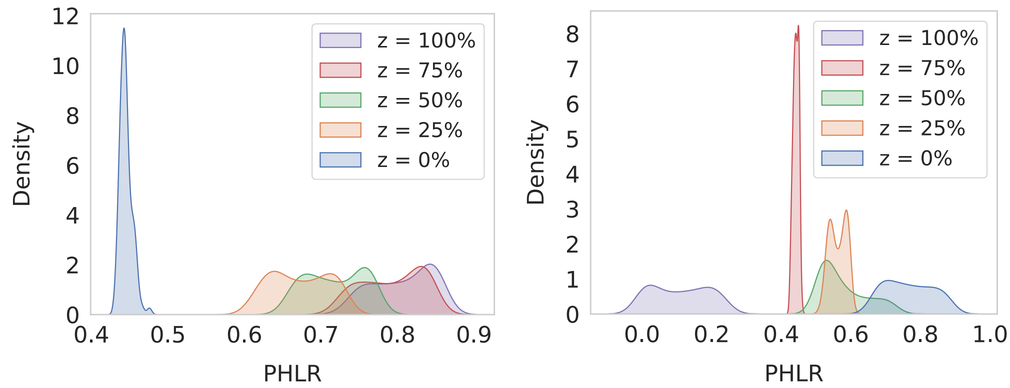
Require: HSI(t)
$\langle w_1, ..., w_n\rangle = t$
$h_1, ..., h_n \leftarrow$ BiLSTM($\langle w_1, ..., w_n\rangle$)
$p_1, ..., p_n \leftarrow$ CRF($h_1, .., h_n$) ▷$\forall_i \quad p_i \in \{B, I, O\}$
$[\langle s_1, e_1\rangle, .., \langle s_k, e_k\rangle] \leftarrow$ GetSpans($p_1, .., p_n$) ▷ *$<s, e>$ are the start and end indices of the hate span.* **return** $[\langle s_1, e_1\rangle, .., \langle s_k, e_k\rangle]$

Require: HIR ($t, \langle s_1, e_1\rangle, .., \langle s_k, e_k\rangle$)
$t' \leftarrow t$
for all $\langle s_i, e_i\rangle$ **do**
▷ *For each identified hate span.*
$H \leftarrow$ BART-encoder (t)
$t'_{[s_i:e_i]} \leftarrow$ BART-decoder ($H, \langle s_i, e_i\rangle$) ▷ *Update hate span with normalised text.*
end forreturn t'

I Temperature probing for anchoring bias

Setup. We simulate multiple annotators under each z value to further corroborate the results. Due to resource constraints, we run this analysis on only `FlanT5-XXL`. We generate 100 output for each sample, by uniformly sampling 100 decoder temperature values ($t \in (0, 2)$).

Findings. For reference, in p_{base}, we obtain a mean (std. dev) percentage of hate label as $0.560(\pm 0.0169)$. While the spread is low for p_{base}, we observe from Figure 4 (a) and (b) that for some z, the spread varies. It is an indicator that decoding temperature can lead to variations in the LLM's predictions when prompted on numerical anchors. We observe more spread on average when using p_{vote} as recorded in Table 16. For p^H_{vote}, with varying temperatures, as the value of z increases, the percentage of hate predicted increases, as evident from the shift towards the right for $z = 100\%$ in Figure 4 (a). The reverse trend is observed for p^N_{vote} in Figure 4 (b), where the curve for $z = 100\%$ is left-shifted, leading to a decrease in the percentage of hate predicted as the majority of non-hateful increases.



(a) `FlanT5-XXL` temperatures (H) (b) `FlanT5-XXL` temperatures (N)

Figure 4: The impact of decoding temperature with varying voting percentages for $p^H_{context}$ on `FlanT5-XXL`.

Table 16: Mean and standard deviation of temperature distribution in Figure 4.

<z>	**Distribution parameters** ($\mu \pm \sigma$)	
	p^H_{vote}	p^N_{vote}
0%	0.45 ± 0.008	0.77 ± 0.067
25%	0.67 ± 0.037	0.56 ± 0.024
50%	0.72 ± 0.037	0.57 ± 0.065
75%	0.79 ± 0.039	0.44 ± 0.008
100%	0.81 ± 0.038	0.11 ± 0.079

J Links to resources

J.1 Datasets and codes

- **Chapter 3**
  - RETINA
  - HateFinetune
  - GotHate
  - DiVA
- **Chapter 4**
  - PolAt
  - FIADD
- **Chapter 5**
  - TOXBART
- **Chapter 6**
  - HateNorm
- **Chapter 7**
  - LLM-Personas

J.2 Blogs

- **To KG or not to KG, that is the question!** ACM SIGMOD Blog
- **Handling Bias in Toxic Speech Detection** Montreal AI Ethics Blog
- **Platform to Track Hate Speech During Elections, Say Most Hateful Tweets From UP** News18 and LiveHindustan

J.3 Shared task

- **Identification of Tokens Contributing to Explicit Hate in Text by Span Detection** Fire HASOC Task 3

REFERENCES

1. **Abercrombie, G.**, **D. Hovy**, and **V. Prabhakaran**, Temporal and second language influence on intra-annotator agreement and stability in hate speech labelling. *In Proc. of the 17th Linguistic Annotation Workshop (LAW-XVII)*. Toronto, Canada, 2023.

2. **Abu Farha, I.**, **S. V. Oprea**, **S. Wilson**, and **W. Magdy**, SemEval-2022 task 6: iSarcasmEval, intended sarcasm detection in English and Arabic. *In Proceedings of the 16th International Workshop on Semantic Evaluation (SemEval-2022)*. Association for Computational Linguistics, Seattle, United States, 2022. URL `https://aclanthology.org/2022.semeval-1.111`.

3. **Adams, C.**, **D. Borkan**, **inversion**, **J. Sorensen**, **L. Dixon**, **L. Vasserman**, and **nithum** (2019). Jigsaw unintended bias in toxicity classification. URL `https://kaggle.com/competitions/jigsaw-unintended-bias-in-toxicity-classification`.

4. **Agarwal, V.**, **Y. Chen**, and **N. Sastry** (2023). Haterephrase: Zero- and few-shot reduction of hate intensity in online posts using large language models. URL `https://arxiv.org/abs/2310.13985`.

5. **Agrawal, S.** and **A. Awekar**, Deep learning for detecting cyberbullying across multiple social media platforms. *In ECIR*. 2018.

6. **Al Kuwatly, H.**, **M. Wich**, and **G. Groh**, Identifying and measuring annotator bias based on annotators' demographic characteristics. *In Proceedings of the Fourth Workshop on Online Abuse and Harms*. 2020.

7. **Alfina, I.**, **R. Mulia**, **M. I. Fanany**, and **Y. Ekanata**, Hate speech detection in the indonesian language: A dataset and preliminary study. *In 2017 International Conference on Advanced Computer Science and Information Systems (ICACSIS)*. 2017.

8. **AlKhamissi, B.**, **F. Ladhak**, **S. Iyer**, **V. Stoyanov**, **Z. Kozareva**, **X. Li**, **P. Fung**, **L. Mathias**, **A. Celikyilmaz**, and **M. Diab**, ToKen: Task decomposition and knowledge infusion for few-shot hate speech detection. *In Proc. of the 2022 Conference on EMNLP*. ACL, Abu Dhabi, United Arab Emirates, 2022.

9. **Alkomah, F.** and **X. Ma** (2022). A literature review of textual hate speech detection methods and datasets. *Information*, **13**(6). ISSN 2078-2489. URL `https://www.mdpi.com/2078-2489/13/6/273`.

10. **Alshaalan, R.** and **H. Al-Khalifa**, Hate speech detection in saudi twittersphere: A deep learning approach. *In Proceedings of the Fifth Arabic Natural Language Processing Workshop*. Association for Computational Linguistics, Barcelona, Spain (Online), 2020. URL `https://aclanthology.org/2020.wanlp-1.2`.

11. **An, S.**, **Y. Li**, **Z. Lin**, **Q. Liu**, **B. Chen**, **Q. Fu**, **W. Chen**, **N. Zheng**, and **J.-G. Lou** (2022). Input-tuning: Adapting unfamiliar inputs to frozen pretrained models.

12. **Antoniak, M.**, **J. Mire**, **M. Sap**, **E. Ash**, and **A. Piper**, Where do people tell stories online? story detection across online communities. *In Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, Bangkok, Thailand, 2024. URL https://aclanthology.org/2024.acl-long.383.

13. **Antypas, D.** and **J. Camacho-Collados**, Robust hate speech detection in social media: A cross-dataset empirical evaluation. *In The 7th Workshop on Online Abuse and Harms (WOAH)*. Association for Computational Linguistics, Toronto, Canada, 2023. URL https://aclanthology.org/2023.woah-1.25.

14. **Arango, A.**, **J. Pérez**, and **B. Poblete**, Hate speech detection is not as easy as you may think: A closer look at model validation. *In Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR'19. Association for Computing Machinery, New York, NY, USA, 2019. ISBN 9781450361729. URL https://doi.org/10.1145/3331184.3331262.

15. **Aroyo, L.**, **L. Dixon**, **N. Thain**, **O. Redfield**, and **R. Rosen**, Crowdsourcing subjective tasks: The case study of understanding toxicity in online discussions. *In Companion Proceedings of The 2019 World Wide Web Conference*. Association for Computing Machinery, New York, NY, USA, 2019*a*.

16. **Aroyo, L.**, **A. Dumitrache**, **O. Inel**, **Z. Szlávik**, **B. Timmermans**, and **C. Welty**, Crowdsourcing inclusivity: Dealing with diversity of opinions, perspectives and ambiguity in annotated data. *In Companion Proceedings of The 2019 World Wide Web Conference*. Association for Computing Machinery, New York, NY, USA, 2019*b*. ISBN 9781450366755.

17. **Arsht, A.** and **D. Etcovitch** (2018). The human cost of online content moderation. *Harvard Journal of Law and Technology*, **2**.

18. **Assenmacher, D.**, **M. Niemann**, **K. Müller**, **M. Seiler**, **D. Riehle**, **H. Trautmann**, and **H. Trautmann**, Rp-mod & rp-crowd: Moderator- and crowd-annotated german news comment datasets. *In Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*, volume 1. Curran, 2021. URL https://datasets-benchmarks-proceedings.neurips.cc/paper_files/paper/2021/file/c9e1074f5b3f9fc8ea15d152add07294-Paper-round2.pdf.

19. **Awal, M. R.**, **R. Cao**, **R. K.-W. Lee**, and **S. Mitrović**, Angrybert: Joint learning target and emotion for hate speech detection. *In Advances in Knowledge Discovery and Data Mining*. Springer International Publishing, Cham, 2021. ISBN 978-3-030-75762-5.

20. **Badjatiya, P.**, **M. Gupta**, and **V. Varma**, Stereotypical bias removal for hate speech detection task using knowledge-based generalizations. *In The World Wide Web Conference*, WWW '19. Association for Computing Machinery, New York, NY, USA, 2019. ISBN 9781450366748. URL https://doi.org/10.1145/3308558.3313504.

21. **Badjatiya, P.**, **S. Gupta**, **M. Gupta**, and **V. Varma**, Deep learning for hate speech detection in tweets. *In Proceedings of the 26th International Conference on World Wide Web Companion*, WWW '17 Companion. International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, CHE, 2017. ISBN 9781450349147. URL https://doi.org/10.1145/3041021.3054223.

22. **Balayn, A.**, **J. Yang**, **Z. Szlavik**, and **A. Bozzon** (2021). Automatic identification of harmful, aggressive, abusive, and offensive language on the web: A survey of technical biases informed by psychology literature. *Trans. Soc. Comput.*, **4**(3). URL https://doi.org/10.1145/3479158.

23. **Balkir, E.**, **I. Nejadgholi**, **K. Fraser**, and **S. Kiritchenko**, Necessity and sufficiency for explaining text classifiers: A case study in hate speech detection. *In Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, Seattle, United States, 2022. URL https://aclanthology.org/2022.naacl-main.192.

24. **Banko, M.**, **B. MacKeen**, and **L. Ray**, A unified taxonomy of harmful content. *In Proceedings of the Fourth Workshop on Online Abuse and Harms*. Association for Computational Linguistics, Online, 2020. URL https://aclanthology.org/2020.alw-1.16.

25. **Basile, V.**, **C. Bosco**, **E. Fersini**, **D. Nozza**, **V. Patti**, **F. M. Rangel Pardo**, **P. Rosso**, and **M. Sanguinetti**, SemEval-2019 task 5: Multilingual detection of hate speech against immigrants and women in Twitter. *In Proceedings of the 13th International Workshop on Semantic Evaluation*. Association for Computational Linguistics, Minneapolis, Minnesota, USA, 2019. URL https://aclanthology.org/S19-2007.

26. **Besta, M.**, **N. Blach**, **A. Kubicek**, **R. Gerstenberger**, **M. Podstawski**, **L. Gianinazzi**, **J. Gajda**, **T. Lehmann**, **H. Niewiadomski**, **P. Nyczyk**, and **T. Hoefler** (2024). Graph of thoughts: Solving elaborate problems with large language models. *Proceedings of the AAAI Conference on Artificial Intelligence*, **38**(16), 17682–17690. ISSN 2159-5399. URL http://dx.doi.org/10.1609/aaai.v38i16.29720.

27. **Bhattacharyya, P.**, *Machine Translation*. Chapman and Hall/CRC, 2015. ISBN 9780429086298.

28. **Bhuiyan, M. M.**, **K. Zhang**, **K. Vick**, **M. A. Horning**, and **T. Mitra**, Feedreflect: A tool for nudging users to assess news credibility on twitter. *In CSCW*. 2018.

29. **Bhutani, M.**, **K. Robinson**, **V. Prabhakaran**, **S. Dave**, and **S. Dev**, SeeGULL multilingual: a dataset of geo-culturally situated stereotypes. *In Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Association for Computational Linguistics, Bangkok, Thailand, 2024. URL https://aclanthology.org/2024.acl-short.75.

30. **Biderman, S.**, **H. Schoelkopf**, **Q. Anthony**, **H. Bradley**, **K. O'Brien**, **E. Hallahan**, **M. A. Khan**, **S. Purohit**, **U. S. Prashanth**, **E. Raff**, **A. Skowron**, **L. Sutawika**, and **O. van der Wal** (2023). Pythia: A suite for analyzing large language models across training and scaling.

31. **Biester, L.**, **V. Sharma**, **A. Kazemi**, **N. Deng**, **S. Wilson**, and **R. Mihalcea**, Analyzing the effects of annotator gender across NLP tasks. *In Proc. of the 1st Workshop on Perspectivist Approaches to NLP @LREC2022*. European Language Resources Association, Marseille, France, 2022. URL https://aclanthology.org/2022.nlperspectives-1.2.

32. **Blodgett, S. L.**, **S. Barocas**, **H. Daumé III**, and **H. Wallach**, Language (technology) is power: A critical survey of "bias" in nlp. *In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. 2020.

33. **Blodgett, S. L.**, **L. Green**, and **B. O'Connor**, Demographic dialectal variation in social media: A case study of African-American English. *In Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Austin, Texas, 2016. URL https://www.aclweb.org/anthology/D16-1120.

34. **Bohra, A.**, **D. Vijay**, **V. Singh**, **S. S. Akhtar**, and **M. Shrivastava**, A dataset of Hindi-English code-mixed social media text for hate speech detection. *In Proceedings of the Second Workshop on Computational Modeling of People's Opinions, Personality, and Emotions in Social Media*. Association for Computational Linguistics, New Orleans, Louisiana, USA, 2018. URL https://aclanthology.org/W18-1105.

35. **Bolukbasi, T.**, **K.-W. Chang**, **J. Y. Zou**, **V. Saligrama**, and **A. T. Kalai**, Man is to computer programmer as woman is to homemaker? debiasing word embeddings. *In Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc., 2016. URL https://proceedings.neurips.cc/paper_files/paper/2016/file/a486cd07e4ac3d270571622f4f316ec5-Paper.pdf.

36. **Botelho, A.**, **S. Hale**, and **B. Vidgen**, Deciphering implicit hate: Evaluating automated detection algorithms for multimodal hate. *In Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*. Association for Computational Linguistics, Online, 2021. URL https://aclanthology.org/2021.findings-acl.166.

37. **Bourigault, S.**, **S. Lamprier**, and **P. Gallinari**, Representation learning for information diffusion through social networks: An embedded cascade model. *In WSDM*. 2016.

38. **Braylan, A.** and **M. Lease**, Modeling and aggregation of complex annotations via annotation distances. *In Proc. of The Web Conference 2020*, WWW '20. Association for Computing Machinery, New York, NY, USA, 2020. ISBN 9781450370233. URL https://doi.org/10.1145/3366423.3380250.

39. **Breitfeller, L.**, **E. Ahn**, **D. Jurgens**, and **Y. Tsvetkov**, Finding microaggressions in the wild: A case for locating elusive phenomena in social media posts. *In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Association for Computational Linguistics, Hong Kong, China, 2019. URL https://aclanthology.org/D19-1176.

40. **Brown, T.**, **B. Mann**, **N. Ryder**, **M. Subbiah**, **J. D. Kaplan**, **P. Dhariwal**, **A. Neelakantan**, **P. Shyam**, **G. Sastry**, **A. Askell**, **S. Agarwal**, **A. Herbert-Voss**, **G. Krueger**, **T. Henighan**, **R. Child**, **A. Ramesh**, **D. Ziegler**, **J. Wu**, **C. Winter**, **C. Hesse**, **M. Chen**, **E. Sigler**, **M. Litwin**, **S. Gray**, **B. Chess**, **J. Clark**, **C. Berner**, **S. McCandlish**, **A. Radford**, **I. Sutskever**, and **D. Amodei**, Language models are few-shot learners. *In Advances in Neural Information Processing Systems*, volume 33. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf.

41. **Calabrese, A.**, **L. Neves**, **N. Shah**, **M. Bos**, **B. Ross**, **M. Lapata**, and **F. Barbieri**, Explainability and hate speech: Structured explanations make social media moderators

faster. *In Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Association for Computational Linguistics, Bangkok, Thailand, 2024. URL https://aclanthology.org/2024.acl-short.38/.

42. **Caliskan, A.**, **J. J. Bryson**, and **A. Narayanan** (2017). Semantics derived automatically from language corpora contain human-like biases. *Science*, **356**(6334), 183–186.

43. **Cao, Q.**, **H. Shen**, **K. Cen**, **W. Ouyang**, and **X. Cheng**, Deephawkes: Bridging the gap between prediction and understanding of information cascades. *In CIKM*. 2017.

44. **Cao, R.** and **R. K.-W. Lee**, HateGAN: Adversarial generative-based data augmentation for hate speech detection. *In Coling*. 2020.

45. **Cao, R.**, **R. K.-W. Lee**, and **T.-A. Hoang**, Deephate: Hate speech detection via multi-faceted text representations. *In 12th ACM Conference on Web Science*, WebSci '20. Association for Computing Machinery, New York, NY, USA, 2020. ISBN 9781450379892. URL https://doi.org/10.1145/3394231.3397890.

46. **Cao, Y. T.**, **Y. Pruksachatkun**, **K.-W. Chang**, **R. Gupta**, **V. Kumar**, **J. Dhamala**, and **A. Galstyan**, On the intrinsic and extrinsic fairness evaluation metrics for contextualized language representations. *In Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Association for Computational Linguistics, Dublin, Ireland, 2022. URL https://aclanthology.org/2022.acl-short.62.

47. **Caselli, T.**, **V. Basile**, **J. Mitrović**, and **M. Granitzer**, HateBERT: Retraining BERT for abusive language detection in English. *In Proceedings of the 5th Workshop on Online Abuse and Harms (WOAH 2021)*. Association for Computational Linguistics, Online, 2021*a*. URL https://aclanthology.org/2021.woah-1.3.

48. **Caselli, T.**, **V. Basile**, **J. Mitrović**, **I. Kartoziya**, and **M. Granitzer**, I feel offended, don't be abusive! implicit/explicit messages in offensive and abusive language. *In Proceedings of the Twelfth Language Resources and Evaluation Conference*. European Language Resources Association, Marseille, France, 2020. ISBN 979-10-95546-34-4. URL https://aclanthology.org/2020.lrec-1.760.

49. **Caselli, T.**, **A. Schelhaas**, **M. Weultjes**, **F. Leistra**, **H. van der Veen**, **G. Timmerman**, and **M. Nissim**, DALC: the Dutch abusive language corpus. *In Proceedings of the 5th Workshop on Online Abuse and Harms (WOAH 2021)*. Association for Computational Linguistics, Online, 2021*b*. URL https://aclanthology.org/2021.woah-1.6.

50. **Celadin, T.**, **F. Panizza**, and **V. Capraro** (2024). Promoting civil discourse on social media using nudges: A tournament of seven interventions. *PNAS Nexus*, **3**(10), pgae380. ISSN 2752-6542. URL https://doi.org/10.1093/pnasnexus/pgae380.

51. **Chakraborty, T.** and **S. Masud** (2022). Nipping in the bud: Detection, diffusion and mitigation of hate speech on social media. *SIGWEB Newsl.*, (Winter). ISSN 1931-1745. URL https://doi.org/10.1145/3522598.3522601.

52. **Chakraborty, T.** and **S. Masud** (2023). Judging the creative prowess of ai. *Nature Machine Intelligence*, **5**(6), 558–558. ISSN 2522-5839. URL https://doi.org/10.1038/s42256-023-00664-y.

53. **Chakraborty, T.** and **S. Masud** (2024). The promethean dilemma of ai at the intersection of hallucination and creativity. *Commun. ACM*, **67**(10), 26–28. ISSN 0001-0782. URL https://doi.org/10.1145/3652102.

54. **Chan, T. K.**, **C. M. Cheung**, and **Z. W. Lee** (2021). Cyberbullying on social networking sites: A literature review and future research directions. *Information & Management*, **58**(2), 103411.

55. **Chang, T.-Y.**, **Y. Liu**, **K. Gopalakrishnan**, **B. Hedayatnia**, **P. Zhou**, and **D. Hakkani-Tur**, Incorporating commonsense knowledge graph in pretrained models for social commonsense tasks. *In Proceedings of Deep Learning Inside Out (DeeLIO): The First Workshop on Knowledge Extraction and Integration for Deep Learning Architectures*. Association for Computational Linguistics, Online, 2020. URL https://aclanthology.org/2020.deelio-1.9.

56. **Chatzakou, D.**, **N. Kourtellis**, **J. Blackburn**, **E. De Cristofaro**, **G. Stringhini**, and **A. Vakali**, Measuring gamergate: A tale of hate, sexism, and bullying. WWW '17 Companion. International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, CHE, 2017. ISBN 9781450349147. URL https://doi.org/10.1145/3041021.3053890.

57. **Chauchard, S.** and **K. Garimella** (2022). What circulates on partisan whatsapp in india? insights from an unusual dataset. *Journal of Quantitative Description: Digital Media*, **2**. URL https://journalqd.org/article/view/2690.

58. **Chaudhary, M.**, **C. Saxena**, and **H. Meng** (2021). Countering online hate speech: An nlp perspective.

59. **Chen, W.**, **X. Chen**, **J. Zhang**, and **K. Huang**, Beyond triplet loss: A deep quadruplet network for person re-identification. *In 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE Computer Society, Los Alamitos, CA, USA, 2017. ISSN 1063-6919. URL https://doi.ieeecomputersociety.org/10.1109/CVPR.2017.145.

60. **Cheng, J.**, **L. Adamic**, **P. A. Dow**, **J. M. Kleinberg**, and **J. Leskovec**, Can cascades be predicted? *In WWW*. 2014.

61. **Chi, Z.**, **S. Huang**, **L. Dong**, **S. Ma**, **B. Zheng**, **S. Singhal**, **P. Bajaj**, **X. Song**, **X.-L. Mao**, **H. Huang**, and **F. Wei**, XLM-E: Cross-lingual language model pre-training via ELECTRA. *In Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, Dublin, Ireland, 2022. URL https://aclanthology.org/2022.acl-long.427.

62. **Chiril, P.**, **E. W. Pamungkas**, **F. Benamara**, **V. Moriceau**, and **V. Patti** (2022). Emotionally informed hate speech detection: A multi-target perspective. *Cognitive Computation*, **14**(1), 322–352. ISSN 1866-9964. URL https://doi.org/10.1007/s12559-021-09862-5.

63. **Chiu, K.-L.**, **A. Collins**, and **R. Alexander** (2022). Detecting hate speech with gpt-3.

64. **Chopra, S.**, **R. Hadsell**, and **Y. LeCun**, Learning a similarity metric discriminatively, with application to face verification. *In 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 1. 2005.

65. **Chopra, S.**, **R. Sawhney**, **P. Mathur**, and **R. Ratn Shah** (2020). Hindi-english hate speech detection: Author profiling, debiasing, and practical perspectives. *Proceedings of the AAAI Conference on Artificial Intelligence*, **34**(01), 386–393. URL https://ojs.aaai.org/index.php/AAAI/article/view/5374.

66. **Chung, H. W.**, **L. Hou**, **S. Longpre**, **B. Zoph**, **Y. Tay**, **W. Fedus**, **Y. Li**, **X. Wang**, **M. Dehghani**, **S. Brahma**, **A. Webson**, **S. S. Gu**, **Z. Dai**, **M. Suzgun**, **X. Chen**, **A. Chowdhery**, **A. Castro-Ros**, **M. Pellat**, **K. Robinson**, **D. Valter**, **S. Narang**, **G. Mishra**, **A. Yu**, **V. Zhao**, **Y. Huang**, **A. Dai**, **H. Yu**, **S. Petrov**, **E. H. Chi**, **J. Dean**, **J. Devlin**, **A. Roberts**, **D. Zhou**, **Q. V. Le**, and **J. Wei** (2022). Scaling instruction-finetuned language models.

67. **Chung, Y.-L.**, **E. Kuzmenko**, **S. S. Tekiroglu**, and **M. Guerini**, CONAN - COunter NArratives through nichesourcing: a multilingual dataset of responses to fight online hate speech. *In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, Florence, Italy, 2019. URL https://aclanthology.org/P19-1271.

68. **Cohen, J.** (1960). A coefficient of agreement for nominal scales. *Educational and Psychological Measurement*, **20**(1), 37–46. URL https://doi.org/10.1177/001316446002000104.

69. **Dahiya, S.**, **S. Sharma**, **D. Sahnan**, **V. Goel**, **E. Chouzenoux**, **V. Elvira**, **A. Majumdar**, **A. Bandhakavi**, and **T. Chakraborty**, Would your tweet invoke hate on the fly? forecasting hate intensity of reply threads on twitter. *In Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, KDD '21. Association for Computing Machinery, New York, NY, USA, 2021. ISBN 9781450383325. URL https://doi.org/10.1145/3447548.3467150.

70. **Dai, N.**, **J. Liang**, **X. Qiu**, and **X. Huang**, Style transformer: Unpaired text style transfer without disentangled latent representation. *In ACL*. 2019.

71. **Davani, A.**, **S. Dev**, **H. Pérez-Urbina**, and **V. Prabhakaran** (2025). A comprehensive framework to operationalize social stereotypes for responsible ai evaluations. URL https://arxiv.org/abs/2501.02074.

72. **Davidson, T.** and **D. Bhattacharya** (2020). Examining racial bias in an online abuse corpus with structural topic modeling. *arXiv preprint arXiv:2005.13041*.

73. **Davidson, T.**, **D. Bhattacharya**, and **I. Weber**, Racial bias in hate speech and abusive language detection datasets. *In Proc. of the Third Workshop on Abusive Language Online*. ACL, Florence, Italy, 2019. URL https://aclanthology.org/W19-3504.

74. **Davidson, T.**, **D. Warmsley**, **M. Macy**, and **I. Weber** (2017). Automated hate speech detection and the problem of offensive language. *Proc. of the International AAAI Conference on Web and Social Media*, **11**(1), 512–515. URL https://ojs.aaai.org/index.php/ICWSM/article/view/14955.

75. **De, A.**, **S. Bhattacharya**, and **N. Ganguly**, Demarcating endogenous and exogenous opinion diffusion process on social networks. *In WWW*. 2018.

76. **de Gibert, O.**, **N. Perez**, **A. García-Pablos**, and **M. Cuadros**, Hate speech dataset from a white supremacy forum. *In Proceedings of the 2nd Workshop on Abusive Language Online (ALW2)*. Association for Computational Linguistics, Brussels, Belgium, 2018. URL https://aclanthology.org/W18-5102.

77. **de Vries, W.**, **A. van Cranenburgh**, and **M. Nissim**, What's so special about BERT's layers? a closer look at the NLP pipeline in monolingual and multilingual models. *In Findings of the Association for Computational Linguistics: EMNLP 2020*. Association for Computational Linguistics, Online, 2020. URL https://aclanthology.org/2020.findings-emnlp.389.

78. **Dementieva, D.**, **N. Babakov**, **A. Ronen**, **A. A. Ayele**, **N. Rizwan**, **F. Schneider**, **X. Wang**, **S. M. Yimam**, **D. Moskovskiy**, **E. Stakovskii**, **E. Kaufman**, **A. Elnagar**, **A. Mukherjee**, and **A. Panchenko**, Multilingual and explainable text detoxification with parallel corpora. *In Proceedings of the 31st International Conference on Computational Linguistics*. Association for Computational Linguistics, Abu Dhabi, UAE, 2025. URL https://aclanthology.org/2025.coling-main.535/.

79. **Deshmukh, J.**, **R. P. Subbanarasimha**, **P. Bassin**, **V. S. Bitra**, **S. Srinivasa**, and **A. Sharma**, An interactive simulator for covid-19 trend analysis. *In CODS COMAD*. 2021.

80. **Deshpande, A.**, **D. Ruiter**, **M. Mosbach**, and **D. Klakow**, StereoKG: Data-driven knowledge graph construction for cultural knowledge and stereotypes. *In Proceedings of the Sixth Workshop on Online Abuse and Harms (WOAH)*. Association for Computational Linguistics, Seattle, Washington (Hybrid), 2022. URL https://aclanthology.org/2022.woah-1.7.

81. **Devlin, J.**, **M.-W. Chang**, **K. Lee**, and **K. Toutanova**, 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)*. Association for Computational Linguistics, Minneapolis, Minnesota, 2019. URL https://aclanthology.org/N19-1423.

82. **Diaz, M.**, **R. Amironesei**, **L. Weidinger**, and **I. Gabriel**, Accounting for offensive speech as a practice of resistance. *In Proceedings of the Sixth Workshop on Online Abuse and Harms (WOAH)*. Association for Computational Linguistics, Seattle, Washington (Hybrid), 2022. URL https://aclanthology.org/2022.woah-1.18.

83. **Dixon, L.**, **J. Li**, **J. Sorensen**, **N. Thain**, and **L. Vasserman**, Measuring and mitigating unintended bias in text classification. *In Proceedings of the 2018 AAAI/ACM Conference on AI, Ethics, and Society*, AIES '18. Association for Computing Machinery, New York, NY, USA, 2018. ISBN 9781450360128. URL https://doi.org/10.1145/3278721.3278729.

84. **Durrani, N.**, **H. Sajjad**, **F. Dalvi**, and **F. Alam**, On the transformation of latent space in fine-tuned NLP models. *In Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Abu Dhabi, United Arab Emirates, 2022. URL https://aclanthology.org/2022.emnlp-main.97.

85. **Dutta, H. S.** and **T. Chakraborty** (2022). Blackmarket-driven collusion on online media: A survey. *ACM/IMS Trans. Data Sci.*, **2**(4). ISSN 2691-1922. URL https://doi.org/10.1145/3517931.

86. **Dutta, S.** and **T. Chakraborty** (2023). Thus spake chatgpt. *Communications of the ACM*, **66**(12), 16–19. ISSN 1557-7317. URL http://dx.doi.org/10.1145/3616863.

87. **Dutta, S.**, **S. Masud**, **S. Chakrabarti**, and **T. Chakraborty**, Deep exogenous and endogenous influence combination for social chatter intensity prediction. *In Proceedings of the 26th SIGKDD International Conference on Knowledge Discovery &; Data Mining*, KDD '20. Association for Computing Machinery, New York, NY, USA, 2020. ISBN 9781450379984. URL `https://doi.org/10.1145/3394486.3403251`.

88. **Elazar, Y.**, **N. Kassner**, **S. Ravfogel**, **A. Feder**, **A. Ravichander**, **M. Mosbach**, **Y. Belinkov**, **H. Schütze**, and **Y. Goldberg** (2023). Measuring causal effects of data statistics on language model's 'factual' predictions.

89. **Elhage, N.**, **N. Nanda**, **C. Olsson**, **T. Henighan**, **N. Joseph**, **B. Mann**, **A. Askell**, **Y. Bai**, **A. Chen**, **T. Conerly**, **N. DasSarma**, **D. Drain**, **D. Ganguli**, **Z. Hatfield-Dodds**, **D. Hernandez**, **A. Jones**, **J. Kernion**, **L. Lovitt**, **K. Ndousse**, **D. Amodei**, **T. Brown**, **J. Clark**, **J. Kaplan**, **S. McCandlish**, and **C. Olah** (2021). A mathematical framework for transformer circuits. *Transformer Circuits Thread*. Https://transformer-circuits.pub/2021/framework/index.html.

90. **ElSherief, M.**, **C. Ziems**, **D. Muchlinski**, **V. Anupindi**, **J. Seybolt**, **M. De Choudhury**, and **D. Yang**, Latent hatred: A benchmark for understanding implicit hate speech. *In Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Online and Punta Cana, Dominican Republic, 2021. URL `https://aclanthology.org/2021.emnlp-main.29`.

91. **Ethayarajh, K.**, How contextual are contextualized word representations? Comparing the geometry of BERT, ELMo, and GPT-2 embeddings. *In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Association for Computational Linguistics, Hong Kong, China, 2019. URL `https://aclanthology.org/D19-1006`.

92. **Fehn Unsvåg, E.** and **B. Gambäck**, The effects of user features on Twitter hate speech detection. *In Proceedings of the 2nd Workshop on Abusive Language Online (ALW2)*. Association for Computational Linguistics, Brussels, Belgium, 2018. URL `https://aclanthology.org/W18-5110`.

93. **Feng, Y.**, **X. Chen**, **B. Y. Lin**, **P. Wang**, **J. Yan**, and **X. Ren**, Scalable multi-hop relational reasoning for knowledge-aware question answering. *In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, Online, 2020. URL `https://aclanthology.org/2020.emnlp-main.99`.

94. **Fischer, A.**, **E. Halperin**, **D. Canetti**, and **A. Jasini** (2018). Why we hate. *Emotion Review*, **10**(4), 309–320.

95. **Fleisig, E.**, **R. Abebe**, and **D. Klein**, When the majority is wrong: Modeling annotator disagreement for subjective tasks. *In Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Singapore, 2023. URL `https://aclanthology.org/2023.emnlp-main.415/`.

96. **Florio, K.**, **V. Basile**, **M. Polignano**, **P. Basile**, and **V. Patti** (2020). Time of your hate: The challenge of time in hate speech detection on social media. *Applied Sciences*, **10**(12). ISSN 2076-3417. URL `https://www.mdpi.com/2076-3417/10/12/4180`.

97. **Fortuna, P.** and **S. Nunes** (2018). A survey on automatic detection of hate speech in text. *ACM Comput. Surv.*, **51**(4). ISSN 0360-0300.

98. **Fortuna, P.**, **J. Rocha da Silva**, **J. Soler-Company**, **L. Wanner**, and **S. Nunes**, A hierarchically-labeled Portuguese hate speech dataset. *In Proceedings of the Third Workshop on Abusive Language Online*. ACL, Florence, Italy, 2019.

99. **Fortuna, P.**, **J. Soler**, and **L. Wanner**, Toxic, hateful, offensive or abusive? what are we really classifying? an empirical analysis of hate speech datasets. *In Proceedings of the Twelfth Language Resources and Evaluation Conference*. European Language Resources Association, Marseille, France, 2020. ISBN 979-10-95546-34-4. URL https://aclanthology.org/2020.lrec-1.838.

100. **Fortuna, P.**, **J. Soler-Company**, and **L. Wanner** (2021). How well do hate speech, toxicity, abusive and offensive language classification models generalize across datasets? *Inf. Process. Manage.*, **58**(3). ISSN 0306-4573. URL https://doi.org/10.1016/j.ipm.2021.102524.

101. **Founta, A.**, **C. Djouvas**, **D. Chatzakou**, **I. Leontiadis**, **J. Blackburn**, **G. Stringhini**, **A. Vakali**, **M. Sirivianos**, and **N. Kourtellis** (2018). Large scale crowdsourcing and characterization of twitter abusive behavior. *Proceedings of the International AAAI Conference on Web and Social Media*, **12**(1). URL https://ojs.aaai.org/index.php/ICWSM/article/view/14991.

102. **Founta, A.** and **L. Specia**, A survey of online hate speech through the causal lens. *In Proceedings of the First Workshop on Causal Inference and NLP*. Association for Computational Linguistics, Punta Cana, Dominican Republic, 2021. URL https://aclanthology.org/2021.cinlp-1.6.

103. **Founta, A. M.**, **D. Chatzakou**, **N. Kourtellis**, **J. Blackburn**, **A. Vakali**, and **I. Leontiadis**, A unified deep learning architecture for abuse detection. *In Proceedings of the 10th ACM Conference on Web Science*, WebSci '19. Association for Computing Machinery, New York, NY, USA, 2019. ISBN 9781450362023. URL https://doi.org/10.1145/3292522.3326028.

104. **Franceschelli, G.** and **M. Musolesi** (2021). Creativity and machine learning: A survey. URL https://arxiv.org/abs/2104.02726.

105. **Gao, L.** and **R. Huang**, Detecting online hate speech using context aware models. *In Proceedings of the International Conference Recent Advances in Natural Language Processing, RANLP 2017*. INCOMA Ltd., Varna, Bulgaria, 2017. URL https://doi.org/10.26615/978-954-452-049-6_036.

106. **Gao, L.**, **A. Kuppersmith**, and **R. Huang**, 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)*. Asian Federation of Natural Language Processing, Taipei, Taiwan, 2017*a*. URL https://aclanthology.org/I17-1078.

107. **Gao, S.**, **H. Pang**, **P. Gallinari**, **J. Guo**, and **N. Kato** (2017*b*). A novel embedding method for information diffusion prediction in social network big data. *IEEE Transactions on Industrial Informatics*, **13**, 2097–2105.

108. **García Benítez-D'Ávila, H.**, Populism and polarization in the digital arena: categorizing and measuring political attacks on twitter. 2022.

109. **Garg, T.**, **S. Masud**, **T. Suresh**, and **T. Chakraborty**, Handling bias in toxic speech detection: A survey. *In ACM Computing Surveys*. Association for Computing Machinery, New York, NY, USA, 2023. ISSN 0360-0300. URL https://doi.org/10.1145/3580494.

110. **Garimella, K.** and **S. Chauchard** (2024). How prevalent is ai misinformation? what our studies in india show so far. *Nature*, **630**(8015), 32–34. ISSN 1476-4687. URL http://dx.doi.org/10.1038/d41586-024-01588-2.

111. **Garrido-Muñoz, I.**, **A. Montejo-Ráez**, **F. Martínez-Santiago**, and **L. A. Ureña-López** (2021). A survey on bias in deep NLP. *Applied Sciences*, **11**(7), 3184. URL https://doi.org/10.3390/app11073184.

112. **Ge, Y.**, **W. Hua**, **K. Mei**, **j. ji**, **J. Tan**, **S. Xu**, **Z. Li**, and **Y. Zhang**, Openagi: When llm meets domain experts. *In Advances in Neural Information Processing Systems*, volume 36. Curran Associates, Inc., 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/1190733f217404edc8a7f4e15a57f301-Paper-Datasets_and_Benchmarks.pdf.

113. **Gehman, S.**, **S. Gururangan**, **M. Sap**, **Y. Choi**, and **N. A. Smith**, RealToxicityPrompts: Evaluating neural toxic degeneration in language models. *In Findings of the Association for Computational Linguistics: EMNLP 2020*. Association for Computational Linguistics, Online, 2020. URL https://aclanthology.org/2020.findings-emnlp.301.

114. **Geva, M.**, **A. Caciularu**, **G. Dar**, **P. Roit**, **S. Sadde**, **M. Shlain**, **B. Tamir**, and **Y. Goldberg**, LM-debugger: An interactive tool for inspection and intervention in transformer-based language models. *In Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*. Association for Computational Linguistics, Abu Dhabi, UAE, 2022. URL https://aclanthology.org/2022.emnlp-demos.2.

115. **Ghosh, K.** and **D. A. Senapati**, Hate speech detection: a comparison of mono and multilingual transformer model with cross-language evaluation. *In Proceedings of the 36th Pacific Asia Conference on Language, Information and Computation*. De La Salle University, Manila, Philippines, 2022. URL https://aclanthology.org/2022.paclic-1.94.

116. **Ghosh, S.**, **M. Suri**, **P. Chiniya**, **U. Tyagi**, **S. Kumar**, and **D. Manocha**, CoSyn: Detecting implicit hate speech in online conversations using a context synergized hyperbolic network. *In Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Singapore, 2023. URL https://aclanthology.org/2023.emnlp-main.377.

117. **Ghosh Chowdhury, A.**, **A. Didolkar**, **R. Sawhney**, and **R. R. Shah**, ARHNet - leveraging community interaction for detection of religious hate speech in Arabic. *In ACL-SRW*. 2019.

118. **Glăveanu, V. P.** (2013). Rewriting the language of creativity: The five a's framework. *Review of General Psychology*, **17**(1), 69–81. URL https://doi.org/10.1037/a0029528.

119. **Goel, V.**, **D. Sahnan**, **S. Dutta**, **A. Bandhakavi**, and **T. Chakraborty** (2023). Hatemongers ride on echo chambers to escalate hate speech diffusion. *PNAS Nexus*, **2**(3), pgad041. ISSN 2752-6542. URL https://doi.org/10.1093/pnasnexus/pgad041.

120. **Gomez, R.**, **J. Gibert**, **L. Gomez**, and **D. Karatzas**, Exploring hate speech detection in multimodal publications. *In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*. 2020.

121. **Gordon, M. L.**, **K. Zhou**, **K. Patel**, **T. Hashimoto**, and **M. S. Bernstein**, The disagreement deconvolution: Bringing machine learning performance metrics in line with reality. *In Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, CHI '21. Association for Computing Machinery, New York, NY, USA, 2021. ISBN 9781450380966. URL https://doi.org/10.1145/3411764.3445423.

122. **Gretton, A.**, **K. M. Borgwardt**, **M. J. Rasch**, **B. Schölkopf**, and **A. Smola** (2012). A kernel two-sample test. *Journal of Machine Learning Research*, **13**(25), 723–773. URL http://jmlr.org/papers/v13/gretton12a.html.

123. **Grover, A.** and **J. Leskovec**, Node2vec: Scalable feature learning for networks. *In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16. Association for Computing Machinery, New York, NY, USA, 2016. ISBN 9781450342322. URL https://doi.org/10.1145/2939672.2939754.

124. **Guan, J.**, **F. Huang**, **Z. Zhao**, **X. Zhu**, and **M. Huang** (2020). A Knowledge-Enhanced Pretraining Model for Commonsense Story Generation. *Transactions of the Association for Computational Linguistics*, **8**, 93–108. ISSN 2307-387X. URL https://doi.org/10.1162/tacl_a_00302.

125. **Guerreiro, N. M.**, **E. Voita**, and **A. Martins**, Looking for a needle in a haystack: A comprehensive study of hallucinations in neural machine translation. *In Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*. Association for Computational Linguistics, Dubrovnik, Croatia, 2023. URL https://aclanthology.org/2023.eacl-main.75.

126. **Guo, K.**, **A. Hu**, **J. Mu**, **Z. Shi**, **Z. Zhao**, **N. Vishwamitra**, and **H. Hu**, An investigation of large language models for real-world hate speech detection. *In 2023 International Conference on Machine Learning and Applications (ICMLA)*. 2023.

127. **Gupta, R.**, **S. Desai**, **M. Goel**, **A. Bandhakavi**, **T. Chakraborty**, and **M. S. Akhtar**, Counterspeeches up my sleeve! intent distribution learning and persistent fusion for intent-conditioned counterspeech generation. *In Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, Toronto, Canada, 2023. URL https://aclanthology.org/2023.acl-long.318.

128. **Hagberg, A.**, **P. Swart**, and **D. S Chult** (2008). Exploring network structure, dynamics, and function using networkx. Technical report, Los Alamos National Lab.(LANL), Los Alamos, NM (United States).

129. **Hamborg, F.**, **N. Meuschke**, **C. Breitinger**, and **B. Gipp**, news-please: A generic news crawler and extractor. *In ICIS*. 2017.

130. **Hamilton, W. L.**, **R. Ying**, and **J. Leskovec**, Inductive representation learning on large graphs. *In Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17. Curran Associates Inc., Red Hook, NY, USA, 2017. ISBN 9781510860964.

131. **Hangartner, D.**, **G. Gennaro**, **S. Alasiri**, **N. Bahrich**, **A. Bornhoft**, **J. Boucher**, **B. B. Demirci**, **L. Derksen**, **A. Hall**, **M. Jochum**, **M. M. Munoz**, **M. Richter**, **F. Vogel**, **S. Wittwer**, **F. Wüthrich**, **F. Gilardi**, and **K. Donnay** (2021). Empathy-based counterspeech can reduce racist hate speech in a social media field experiment. *PNAS*, **118**(50).

132. **Hanu, L.** and **Unitary team** (2020). Detoxify. Github. https://github.com/unitaryai/detoxify.

133. **Harriss, J.** (2015). Hindu nationalism in action: The bharatiya janata party and indian politics. *South Asia: Journal of South Asian Studies*, **38**(4), 712–718. URL https://doi.org/10.1080/00856401.2015.1089826.

134. **Hartvigsen, T.**, **S. Gabriel**, **H. Palangi**, **M. Sap**, **D. Ray**, and **E. Kamar**, ToxiGen: A large-scale machine-generated dataset for adversarial and implicit hate speech detection. *In Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, Dublin, Ireland, 2022. URL https://aclanthology.org/2022.acl-long.234.

135. **He, B.**, **C. Ziems**, **S. Soni**, **N. Ramakrishnan**, **D. Yang**, and **S. Kumar** (2022). Racism is a virus: anti-asian hate and counterspeech in social media during the covid-19 crisis, 90–94. URL https://doi.org/10.1145/3487351.3488324.

136. **He, X.**, **Z. Lin**, **Y. Gong**, **A. Jin**, **H. Zhang**, **C. Lin**, **J. Jiao**, **S. M. Yiu**, **N. Duan**, **W. Chen**, *et al.* (2023). Annollm: Making large language models to be better crowdsourced annotators. *arXiv preprint arXiv:2303.16854*. URL https://arxiv.org/abs/2303.16854.

137. **Hecht-Nielsen, R.**, *Confabulation Theory: The Mechanism of Thought*. Springer Berlin Heidelberg, Berlin, Heidelberg, 2007. ISBN 978-3-540-49605-2. URL https://doi.org/10.1007/978-3-540-49605-2_1.

138. **Hochreiter, S.** and **J. Schmidhuber** (1997). Long short-term memory. *Neural Computation*, **9**(8), 1735–1780. URL https://doi.org/10.1162/neco.1997.9.8.1735.

139. **Hovy, D.** and **D. Yang**, The importance of modeling social factors of language: Theory and practice. *In Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, Online, 2021. URL https://aclanthology.org/2021.naacl-main.49/.

140. **Hu, Y.**, **S. Farnham**, and **K. Talamadupula**, Predicting user engagement on twitter with real-world events. *In ICWSM*. 2015.

141. **Hu, Z.**, **Z. Yang**, **X. Liang**, **R. Salakhutdinov**, and **E. P. Xing**, Toward controlled generation of text. *In Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ICML'17. JMLR.org, 2017.

142. **Huang, F.**, **H. Kwak**, and **J. An**, Chain of explanation: New prompting method to generate quality natural language explanation for implicit hate speech. *In Companion Proceedings of the ACM Web Conference 2023*, WWW '23 Companion. Association for Computing Machinery, New York, NY, USA, 2023*a*. ISBN 9781450394192. URL https://doi.org/10.1145/3543873.3587320.

143. **Huang, F.**, **H. Kwak**, and **J. An**, Is chatgpt better than human annotators? potential and limitations of chatgpt in explaining implicit hate speech. *In Companion Proceedings of the ACM Web Conference 2023*, WWW '23 Companion. Association for Computing Machinery, New York, NY, USA, 2023*b*. ISBN 9781450394192. URL https://doi.org/10.1145/3543873.3587368.

144. **Huang, X.**, **L. Xing**, **F. Dernoncourt**, and **M. J. Paul**, Multilingual Twitter corpus and baselines for evaluating demographic bias in hate speech recognition. *In Proceedings of the Twelfth Language Resources and Evaluation Conference*. European Language Resources Association, Marseille, France, 2020. ISBN 979-10-95546-34-4. URL https://aclanthology.org/2020.lrec-1.180/.

145. **Huszár, F.**, **S. I. Ktena**, **C. O'Brien**, **L. Belli**, **A. Schlaikjer**, and **M. Hardt** (2021). Algorithmic amplification of politics on twitter. *Proceedings of the National Academy of Sciences*, **119**(1), e2025334119. URL https://doi.org/10.1073/pnas.2025334119.

146. **Ibrohim, M. O.** and **I. Budi**, Multi-label hate speech and abusive language detection in Indonesian Twitter. *In Proceedings of the Third Workshop on Abusive Language Online*. Association for Computational Linguistics, Florence, Italy, 2019. URL https://aclanthology.org/W19-3506.

147. **İhtiyar, M.**, **Ö. Özdemir**, **M. Erengül**, and **A. Özgür**, A dataset for investigating the impact of context for offensive language detection in tweets. *In Findings of the Association for Computational Linguistics: EMNLP 2023*. Association for Computational Linguistics, Singapore, 2023. URL https://aclanthology.org/2023.findings-emnlp.103.

148. **Ilan, T.** and **D. Vilenchik**, HARALD: Augmenting hate speech data sets with real data. *In Findings of the Association for Computational Linguistics: EMNLP 2022*. Association for Computational Linguistics, Abu Dhabi, United Arab Emirates, 2022. URL https://aclanthology.org/2022.findings-emnlp.165.

149. **Inan, H.**, **K. Upasani**, **J. Chi**, **R. Rungta**, **K. Iyer**, **Y. Mao**, **M. Tontchev**, **Q. Hu**, **B. Fuller**, **D. Testuggine**, and **M. Khabsa** (2023). Llama guard: Llm-based input-output safeguard for human-ai conversations. URL https://arxiv.org/abs/2312.06674.

150. **Jacomy, M.**, **T. Venturini**, **S. Heymann**, and **M. Bastian** (2014). Forceatlas2, a continuous graph layout algorithm for handy network visualization designed for the gephi software. *PLoS ONE*, **9**(6), e98679. ISSN 1932-6203. URL http://dx.doi.org/10.1371/journal.pone.0098679.

151. **Jafri, F. A.**, **M. A. Siddiqui**, **S. Thapa**, **K. Rauniyar**, **U. Naseem**, and **I. Razzak**, Uncovering the dynamics of political hate speech during indian election campaign: A new low-resource dataset and baselines. *In Workshop Proceedings of the 17th ICWSM*. 2023.

152. **Jakesch, M.**, **K. Garimella**, **D. Eckles**, and **M. Naaman** (2021). Trend alert: A cross-platform organization manipulated twitter trends in the indian general election. *Proc. ACM Hum.-Comput. Interact.*, **5**.

153. **Jenness, S. M.**, **S. M. Goodreau**, and **M. Morris** (2018). Epimodel: An r package for mathematical modeling of infectious disease over networks. *Journal of Statistical Software*, **84**(8), 1–47. URL https://www.jstatsoft.org/index.php/jss/article/view/v084i08.

154. **Jha, A.** and **R. Mamidi**, When does a compliment become sexist? analysis and classification of ambivalent sexism using twitter data. *In Proceedings of the Second Workshop on NLP and Computational Social Science*. Vancouver, Canada, 2017.

155. **Jhaver, S.**, **A. Q. Zhang**, **Q. Z. Chen**, **N. Natarajan**, **R. Wang**, and **A. X. Zhang** (2023). Personalizing content moderation on social media: User perspectives on moderation choices, interface design, and labor. *Proc. ACM Hum.-Comput. Interact.*, **7**(CSCW2). URL https://doi.org/10.1145/3610080.

156. **Jiang, A. Q.**, **A. Sablayrolles**, **A. Mensch**, **C. Bamford**, **D. S. Chaplot**, **D. d. l. Casas**, **F. Bressand**, **G. Lengyel**, **G. Lample**, **L. Saulnier**, *et al.* (2023*a*). Mistral 7b. *arXiv preprint arXiv:2310.06825*. URL https://arxiv.org/abs/2310.06825.

157. **Jiang, J. A.**, **P. Nie**, **J. R. Brubaker**, and **C. Fiesler** (2023*b*). A trade-off-centered framework of content moderation. *ACM Trans. Comput.-Hum. Interact.*, **30**(1). ISSN 1073-0516. URL https://doi.org/10.1145/3534929.

158. **Jin, Y.**, **M. Chandra**, **G. Verma**, **Y. Hu**, **M. De Choudhury**, and **S. Kumar**, Better to ask in english: Cross-lingual evaluation of large language models for healthcare queries. *In Proceedings of the ACM on Web Conference 2024*, WWW '24. Association for Computing Machinery, New York, NY, USA, 2024. ISBN 9798400701719. URL https://doi.org/10.1145/3589334.3645643.

159. **Johnson, N. F.**, **R. Leahy**, **N. J. Restrepo**, **N. Velasquez**, **M. Zheng**, **P. Manrique**, **P. Devkota**, and **S. Wuchty** (2019). Hidden resilience and adaptive dynamics of the global online hate ecology. *Nature*, **573**(7773), 261–265. ISSN 1476-4687. URL https://doi.org/10.1038/s41586-019-1494-7.

160. **Kakwani, D.**, **A. Kunchukuttan**, **S. Golla**, **G. N.C.**, **A. Bhattacharyya**, **M. M. Khapra**, and **P. Kumar**, IndicNLPSuite: Monolingual corpora, evaluation benchmarks and pre-trained multilingual language models for Indian languages. *In Findings of the Association for Computational Linguistics: EMNLP 2020*. Association for Computational Linguistics, Online, 2020. URL https://aclanthology.org/2020.findings-emnlp.445.

161. **Kanclerz, K.**, **M. Gruza**, **K. Karanowski**, **J. Bielaniewicz**, **P. Milkowski**, **J. Kocon**, and **P. Kazienko**, What if ground truth is subjective? personalized deep neural hate speech detection. *In Proc. of the 1st Workshop on Perspectivist Approaches to NLP @LREC2022*. European Language Resources Association, Marseille, France, 2022.

162. **Kapoor, R.**, **Y. Kumar**, **K. Rajput**, **R. R. Shah**, **P. Kumaraguru**, and **R. Zimmermann** (2018). Mind your language: Abuse and offense detection for code-switched languages. *arXiv*, **1809.08652**.

163. **Karim, M. R.**, **S. K. Dey**, **T. Islam**, **S. Sarker**, **M. H. Menon**, **K. Hossain**, **M. A. Hossain**, and **S. Decker**, Deephateexplainer: Explainable hate speech detection in under-resourced bengali language. *In 2021 IEEE 8th International Conference on Data Science and Advanced Analytics (DSAA)*. 2021.

164. **Katsaros, M.**, **K. Yang**, and **L. Fratamico** (2022). Reconsidering tweets: Intervening during tweet creation decreases offensive content. **16**, 477–487. URL https://ojs.aaai.org/index.php/ICWSM/article/view/19308.

165. **Kaufman, G. F.** and **L. K. Libby** (2012). Changing beliefs and behavior through experience-taking. *Journal of personality and social psychology*, **103**(1), 1—19.

166. **Kaufman, J. C.** and **V. P. Glăveanu**, *A Review of Creativity Theories: What Questions Are We Trying to Answer?*. Cambridge Handbooks in Psychology. Cambridge University Press, 2019, 2 edition, 27–43.

167. **Kempe, D.**, **J. Kleinberg**, and **É. Tardos**, Maximizing the spread of influence through a social network. *In SIGKDD*. 2003.

168. **Kennedy, B.**, **M. Atari**, **A. M. Davani**, **L. Yeh**, **A. Omrani**, **Y. Kim**, **K. Coombs**, **S. Havaldar**, **G. Portillo-Wightman**, **E. Gonzalez**, **J. Hoover**, **A. Azatian**, **A. Hussain**, **A. Lara**, **G. Cardenas**, **A. Omary**, **C. Park**, **X. Wang**, **C. Wijaya**, **Y. Zhang**, **B. Meyerowitz**, and **M. Dehghani** (2022). Introducing the gab hate corpus: defining and applying hate-based rhetoric to social media posts at scale. *Lang. Resour. Eval.*, **56**(1), 79–108. ISSN 1574-020X. URL https://doi.org/10.1007/s10579-021-09569-x.

169. **Kennedy, B.**, **X. Jin**, **A. Mostafazadeh Davani**, **M. Dehghani**, and **X. Ren**, Contextualizing hate speech classifiers with post-hoc explanation. *In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, Online, 2020. URL https://www.aclweb.org/anthology/2020.acl-main.483.

170. **Kermack, W. O.** and **A. G. McKendrick** (1927). A contribution to the mathematical theory of epidemics. *Proceedings of the royal society of london. Series A, Containing papers of a mathematical and physical character*, **115**(772), 700–721.

171. **Khosla, A.**, **A. Das Sarma**, and **R. Hamid**, What makes an image popular? *In Proceedings of the 23rd International Conference on World Wide Web*, WWW '14. Association for Computing Machinery, New York, NY, USA, 2014. ISBN 9781450327442. URL https://doi.org/10.1145/2566486.2567996.

172. **Kiela, D.**, **H. Firooz**, **A. Mohan**, **V. Goswami**, **A. Singh**, **P. Ringshia**, and **D. Testuggine**, The hateful memes challenge: Detecting hate speech in multimodal memes. *In Advances in Neural Information Processing Systems*, volume 33. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper/2020/file/1b84c4cee2b8b3d823b30e2d604b1878-Paper.pdf.

173. **Kim, J. Y.**, **C. Ortiz**, **S. Nam**, **S. Santiago**, and **V. Datta** (2020). Intersectional bias in hate speech and abusive language datasets.

174. **Kim, S.** and **G. Lee**, Adversarial DPO: Harnessing harmful data for reducing toxicity with minimal impact on coherence and evasiveness in dialogue agents. *In Findings of the Association for Computational Linguistics: NAACL 2024*. Association for Computational Linguistics, Mexico City, Mexico, 2024. URL https://aclanthology.org/2024.findings-naacl.118/.

175. **Kim, Y.**, Convolutional neural networks for sentence classification. *In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, Doha, Qatar, 2014. URL https://aclanthology.org/D14-1181.

176. **Kim, Y.**, **S. Park**, and **Y.-S. Han**, Generalizable implicit hate speech detection using contrastive learning. *In Proceedings of the 29th International Conference on Computational Linguistics*. International Committee on Computational Linguistics, Gyeongju, Republic of Korea, 2022. URL https://aclanthology.org/2022.coling-1.579.

177. **Kim, Y.**, **S. Park**, **Y. Namgoong**, and **Y.-S. Han**, ConPrompt: Pre-training a language model with machine-generated data for implicit hate speech detection. *In Findings of the Association for Computational Linguistics: EMNLP 2023*. Association for Computational Linguistics, Singapore, 2023. URL https://aclanthology.org/2023.findings-emnlp.731.

178. **Kingma, D. P.** and **J. Ba** (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*. URL https://arxiv.org/abs/1412.6980.

179. **Kipf, T. N.** and **M. Welling**, Semi-supervised classification with graph convolutional networks. *In 5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017. URL https://openreview.net/forum?id=SJU4ayYgl.

180. **Kirk, H.**, **B. Vidgen**, **P. Rottger**, **T. Thrush**, and **S. Hale**, Hatemoji: A test suite and adversarially-generated dataset for benchmarking and detecting emoji-based hate. *In Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, Seattle, United States, 2022. URL https://aclanthology.org/2022.naacl-main.97/.

181. **Kirk, H.**, **W. Yin**, **B. Vidgen**, and **P. Röttger**, SemEval-2023 task 10: Explainable detection of online sexism. *In Proceedings of the 17th International Workshop on Semantic Evaluation (SemEval-2023)*. Association for Computational Linguistics, Toronto, Canada, 2023. URL https://aclanthology.org/2023.semeval-1.305.

182. **Kirtz, J. L.** and **Z. Talat**, Futures for research on hate speech in online social media platforms. *In Challenges and perspectives of hate speech research*, volume 12 of *Digital Communication Research*. Berlin, 2023. ISBN 978-3-945681-12-1, 467–482.

183. **Kojima, T.**, **S. S. Gu**, **M. Reid**, **Y. Matsuo**, and **Y. Iwasawa**, Large language models are zero-shot reasoners. *In Advances in Neural Information Processing Systems*, volume 35. Curran Associates, Inc., 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/8bb0d291acd4acf06ef112099c16f326-Paper-Conference.pdf.

184. **Koufakou, A.**, **E. W. Pamungkas**, **V. Basile**, and **V. Patti**, HurtBERT: Incorporating lexical features with BERT for the detection of abusive language. *In Proceedings of the Fourth Workshop on Online Abuse and Harms*. Association for Computational Linguistics, Online, 2020. URL https://aclanthology.org/2020.alw-1.5.

185. **Kralj Novak, P.**, **T. Scantamburlo**, **A. Pelicon**, **M. Cinelli**, **I. Mozetič**, and **F. Zollo**, Handling disagreement in hate speech modelling. *In Information Processing and Management of Uncertainty in Knowledge-Based Systems*. Springer International Publishing, Cham, 2022. ISBN 978-3-031-08974-9.

186. **Krippendorff, K.**, Computing krippendorff's alpha-reliability. 2011.

187. **Krishna, K.**, **G. S. Tomar**, **A. Parikh**, **N. Papernot**, and **M. Iyyer**, Thieves of sesame street: Model extraction on bert-based apis. 2020.

188. **Krishna, R.**, **D. Lee**, **L. Fei-Fei**, and **M. S. Bernstein** (2022). Socially situated artificial intelligence enables learning from human interaction. *Proceedings of the National Academy of Sciences*, **119**(39). URL https://doi.org/10.1073/pnas.2115730119.

189. **Kruk, J.**, **M. Marchini**, **R. Magu**, **C. Ziems**, **D. Muchlinski**, and **D. Yang**, Silent signals, loud impact: LLMs for word-sense disambiguation of coded dog whistles. *In Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, Bangkok, Thailand, 2024. URL https://aclanthology.org/2024.acl-long.675.

190. **Kulkarni, A.**, **S. Masud**, **V. Goyal**, and **T. Chakraborty**, Revisiting hate speech benchmarks: From data curation to system deployment. *In Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, KDD '23. Association for Computing Machinery, New York, NY, USA, 2023. ISBN 9798400701030. URL https://doi.org/10.1145/3580305.3599896.

191. **Kumar, S.**, **A. Chandrabose**, and **B. R. Chakravarthi**, An overview of fairness in data – illuminating the bias in data pipeline. *In Proceedings of the First Workshop on Language Technology for Equality, Diversity and Inclusion*. Association for Computational Linguistics, Kyiv, 2021. URL https://aclanthology.org/2021.ltedi-1.5.

192. **Kumar, S.**, **A. Kulkarni**, **M. S. Akhtar**, and **T. Chakraborty**, When did you become so smart, oh wise one?! sarcasm explanation in multi-modal multi-party dialogues. *In Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, Dublin, Ireland, 2022. URL https://aclanthology.org/2022.acl-long.411.

193. **Kumar, S.**, **R. S**, **M. Akhtar**, and **T. Chakraborty**, From multilingual complexity to emotional clarity: Leveraging commonsense to unveil emotions in code-mixed dialogues. *In Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Singapore, 2023. URL https://aclanthology.org/2023.emnlp-main.598/.

194. **Laaksonen, S.-M.**, **J. Haapoja**, **T. Kinnunen**, **M. Nelimarkka**, and **R. Pöyhtäri** (2020). The datafication of hate: Expectations and challenges in automated hate speech monitoring. *Frontiers in Big Data*, **3**(3), 1–16.

195. **LAI, M.**, **F. Celli**, **A. Ramponi**, **S. Tonelli**, **C. Bosco**, and **V. Patti**, Haspeede3 at evalita 2023: Overview of the political and religious hate speech detection task. 2023. 8th Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop, EVALITA 2023 ; Conference date: 01-01-2023.

196. **Le, Q.** and **T. Mikolov**, Distributed representations of sentences and documents. *In Proceedings of the 31st International Conference on Machine Learning*, volume 32 of *Proceedings of Machine Learning Research*. PMLR, Bejing, China, 2014. URL `https://proceedings.mlr.press/v32/le14.html`.

197. **Lee, M.** (2023). A mathematical investigation of hallucination and creativity in gpt models. *Mathematics*, **11**(10). ISSN 2227-7390. URL `https://www.mdpi.com/2227-7390/11/10/2320`.

198. **Lee, N.**, **C. Jung**, **J. Myung**, **J. Jin**, **J. Camacho-Collados**, **J. Kim**, and **A. Oh** (2024). Exploring cross-cultural differences in English hate speech annotations: From dataset construction to analysis, 4205–4224. URL `https://aclanthology.org/2024.naacl-long.236/`.

199. **Lewis, M.**, **Y. Liu**, **N. Goyal**, **M. Ghazvininejad**, **A. Mohamed**, **O. Levy**, **V. Stoyanov**, and **L. Zettlemoyer**, BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, Online, 2020. URL `https://aclanthology.org/2020.acl-main.703`.

200. **Li, C.**, **J. Ma**, **X. Guo**, and **Q. Mei**, Deepcas: An end-to-end predictor of information cascades. *In WWW*. 2017.

201. **Li, D.**, **Z. Ren**, and **E. Kanoulas**, Crowdgp: A gaussian process model for inferring relevance from crowd annotations. *In Proc. of the Web Conference 2021*, WWW '21. Association for Computing Machinery, New York, NY, USA, 2021. ISBN 9781450383127. URL `https://doi.org/10.1145/3442381.3450047`.

202. **Li, H.**, **H. Li**, and **S. S. Bhowmick**, Chassis: Conformity meets online information diffusion. *In SIGMOD*. Association for Computing Machinery, New York, NY, USA, 2020. ISBN 9781450367356.

203. **Li, J.**, **R. Jia**, **H. He**, and **P. Liang**, Delete, retrieve, generate: a simple approach to sentiment and style transfer. *In Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*. New Orleans, Louisiana, 2018.

204. **Li, X.**, **Z. X. Yong**, and **S. Bach**, Preference tuning for toxicity mitigation generalizes across languages. *In Findings of the Association for Computational Linguistics: EMNLP 2024*. Association for Computational Linguistics, Miami, Florida, USA, 2024. URL `https://aclanthology.org/2024.findings-emnlp.784/`.

205. **Li, Y.**, **G. Zhang**, **B. Yang**, **C. Lin**, **A. Ragni**, **S. Wang**, and **J. Fu**, HERB: Measuring hierarchical regional bias in pre-trained language models. *In Findings of the ACL: AACL-IJCNLP 2022*. ACL, Online only, 2022. URL `https://aclanthology.org/2022.findings-aacl.32`.

206. **Likert, R.** (1932). A technique for the measurement of attitudes. *Archives of Psychology*, **22 140**, 55–55.

207. **Lima, L.**, **J. C. S. Reis**, **P. Melo**, **F. Murai**, and **F. Benevenuto**, Characterizing (un)moderated textual data in social systems. *In 2020 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*. 2020.

208. **Lin, C.-Y.**, ROUGE: A package for automatic evaluation of summaries. *In Text Summarization Branches Out*. Association for Computational Linguistics, Barcelona, Spain, 2004. URL https://aclanthology.org/W04-1013.

209. **Lin, J.**, Leveraging world knowledge in implicit hate speech detection. *In Proceedings of the Second Workshop on NLP for Positive Impact (NLP4PI)*. Association for Computational Linguistics, Abu Dhabi, United Arab Emirates (Hybrid), 2022. URL https://aclanthology.org/2022.nlp4pi-1.4.

210. **Lin, T.-Y.**, **P. Goyal**, **R. Girshick**, **K. He**, and **P. Dollár**, Focal loss for dense object detection. *In 2017 IEEE International Conference on Computer Vision (ICCV)*. 2017.

211. **Liu, H.**, **D. Tam**, **M. Muqeeth**, **J. Mohta**, **T. Huang**, **M. Bansal**, and **C. A. Raffel**, Few-shot parameter-efficient fine-tuning is better and cheaper than in-context learning. *In Advances in Neural Information Processing Systems*, volume 35. Curran Associates, Inc., 2022*a*. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/0cde695b83bd186c1fd456302888454c-Paper-Conference.pdf.

212. **Liu, J.**, **D. Shen**, **Y. Zhang**, **B. Dolan**, **L. Carin**, and **W. Chen**, What makes good in-context examples for GPT-3? *In Proceedings of Deep Learning Inside Out (DeeLIO 2022): The 3rd Workshop on Knowledge Extraction and Integration for Deep Learning Architectures*. Association for Computational Linguistics, Dublin, Ireland and Online, 2022*b*. URL https://aclanthology.org/2022.deelio-1.10.

213. **Liu, W.**, **Y. Wen**, **Z. Yu**, and **M. Yang**, Large-margin softmax loss for convolutional neural networks. *In Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*. PMLR, New York, New York, USA, 2016. URL https://proceedings.mlr.press/v48/liud16.html.

214. **Liu, Y.**, **M. D. Ma**, **W. Qin**, **A. Zhou**, **J. Chen**, **W. Shi**, **W. Wang**, and **D. Yang**, Decoding susceptibility: Modeling misbelief to misinformation through a computational approach. *In Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Miami, Florida, USA, 2024. URL https://aclanthology.org/2024.emnlp-main.846.

215. **Liu, Y.**, **M. Ott**, **N. Goyal**, **J. Du**, **M. Joshi**, **D. Chen**, **O. Levy**, **M. Lewis**, **L. Zettlemoyer**, and **V. Stoyanov** (2019). Roberta: A robustly optimized bert pretraining approach. URL https://arxiv.org/abs/1907.11692.

216. **Ljubešić, N.**, **I. Mozetič**, and **P. K. Novak** (2022). Quantifying the impact of context on the quality of manual hate speech annotation. *Natural Language Engineering*, 1–14. URL https://doi.org/10.1017/s1351324922000353.

217. **Longpre, S.**, **G. Yauney**, **E. Reif**, **K. Lee**, **A. Roberts**, **B. Zoph**, **D. Zhou**, **J. Wei**, **K. Robinson**, **D. Mimno**, and **D. Ippolito**, A pretrainer's guide to training data: Measuring the effects of data age, domain coverage, quality, & toxicity. *In Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*. Association for Computational Linguistics, Mexico City, Mexico, 2024. URL https://aclanthology.org/2024.naacl-long.179/.

218. **Lu, J.**, **A. Liu**, **F. Dong**, **F. Gu**, **J. Gama**, and **G. Zhang** (2019). Learning under concept drift: A review. *IEEE Transactions on Knowledge and Data Engineering*, **31**(12), 2346–2363.

219. **Ludwig, F.**, **K. Dolos**, **T. Zesch**, and **E. Hobley**, Improving generalization of hate speech detection systems to novel target groups via domain adaptation. *In Proceedings of the Sixth Workshop on Online Abuse and Harms (WOAH)*. Association for Computational Linguistics, Seattle, Washington (Hybrid), 2022. URL https://aclanthology.org/2022.woah-1.4.

220. **Lundberg, S. M.** and **S.-I. Lee**, A unified approach to interpreting model predictions. *In Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17. Curran Associates Inc., Red Hook, NY, USA, 2017. ISBN 9781510860964.

221. **Luo, F.**, **P. Li**, **P. Yang**, **J. Zhou**, **Y. Tan**, **B. Chang**, **Z. Sui**, and **X. Sun**, Towards fine-grained text sentiment transfer. *In ACL*. 2019.

222. **Lupu, Y.**, **R. Sear**, **N. Velásquez**, **R. Leahy**, **N. J. Restrepo**, **B. Goldberg**, and **N. F. Johnson** (2023). Offline events and online hate. *PLOS ONE*, **18**(1), 1–14. URL https://doi.org/10.1371/journal.pone.0278511.

223. **Lyu, X.**, **S. Min**, **I. Beltagy**, **L. Zettlemoyer**, and **H. Hajishirzi**, Z-ICL: Zero-shot in-context learning with pseudo-demonstrations. *In Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, Toronto, Canada, 2023. URL https://aclanthology.org/2023.acl-long.129.

224. **Madaan, A.**, **A. Setlur**, **T. Parekh**, **B. Poczos**, **G. Neubig**, **Y. Yang**, **R. Salakhutdinov**, **A. W. Black**, and **S. Prabhumoye**, Politeness transfer: A tag and generate approach. *In ACL*. 2020.

225. **Mandl, T.**, **S. Modha**, **A. Kumar M**, and **B. R. Chakravarthi**, Overview of the HASOC Track at FIRE 2020: Hate Speech and Offensive Language Identification in Tamil, Malayalam, Hindi, English and German. *In Proc. of the 12th Annual Meeting of the Forum for Information Retrieval Evaluation*. 2020.

226. **Mandl, T.**, **S. Modha**, **P. Majumder**, **D. Patel**, **M. Dave**, **C. Mandlia**, and **A. Patel**, Overview of the hasoc track at fire 2019: Hate speech and offensive content identification in indo-european languages. *In Proceedings of the 11th Forum for Information Retrieval Evaluation*, FIRE '19. Association for Computing Machinery, New York, NY, USA, 2019. ISBN 9781450377508. URL https://doi.org/10.1145/3368567.3368584.

227. **Manerba, M. M.**, **K. Stańczak**, **R. Guidotti**, and **I. Augenstein**, Social Bias Probing: Fairness Benchmarking for Language Models. *In Proceedings of EMNLP*. Association for Computational Linguistics, 2024.

228. **Manzini, A.**, **G. Keeling**, **L. Alberts**, **S. Vallor**, **M. R. Morris**, and **I. Gabriel** (2024). The code that binds us: Navigating the appropriateness of human-ai assistant relationships. *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society*, **7**(1), 943–957. URL https://ojs.aaai.org/index.php/AIES/article/view/31694.

229. **Massanari, A.** (2017). #gamergate and the fappening: How reddit's algorithm, governance, and culture support toxic technocultures. *New Media & Society*, **19**(3), 329–346. URL https://doi.org/10.1177/1461444815608807.

230. **Masud, S.**, **A. Bajpai**, and **T. Chakraborty** (2024*a*). Focal inferential infusion coupled with tractable density discrimination for implicit hate detection. *Natural Language Processing*, 1–27.

231. **Masud, S.**, **M. Bedi**, **M. A. Khan**, **M. S. Akhtar**, and **T. Chakraborty**, Proactively reducing the hate intensity of online posts via hate speech normalization. *In Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, KDD '22. Association for Computing Machinery, New York, NY, USA, 2022. ISBN 9781450393850. URL https://doi.org/10.1145/3534678.3539161.

232. **Masud, S.** and **T. Chakraborty**, Political mud slandering and power dynamics during indian assembly elections. 1. 2023. ISSN 1869-5469. URL https://doi.org/10.1007/s13278-023-01103-x.

233. **Masud, S.**, **S. Dutta**, **S. Makkar**, **C. Jain**, **V. Goyal**, **A. Das**, and **T. Chakraborty**, Hate is the new infodemic: A topic-aware modeling of hate speech diffusion on twitter. *In 2021 IEEE 37th International Conference on Data Engineering (ICDE)*. 2021.

234. **Masud, S.**, **M. A. Khan**, **V. Goyal**, **M. S. Akhtar**, and **T. Chakraborty**, Probing critical learning dynamics of PLMs for hate speech detection. *In Findings of the Association for Computational Linguistics: EACL 2024*. Association for Computational Linguistics, St. Julian's, Malta, 2024*b*. URL https://aclanthology.org/2024.findings-eacl.55.

235. **Masud, S.**, **S. Singh**, **V. Hangya**, **A. Fraser**, and **T. Chakraborty** (2024*c*). Hate personified: Investigating the role of llms in content moderation. URL https://arxiv.org/abs/2410.02657.

236. **Mathew, B.**, **R. Dutt**, **P. Goyal**, and **A. Mukherjee**, Spread of hate speech in online social media. *In Proceedings of the 10th ACM Conference on Web Science*, WebSci '19. Association for Computing Machinery, New York, NY, USA, 2019. ISBN 9781450362023. URL https://doi.org/10.1145/3292522.3326034.

237. **Mathew, B.**, **A. Illendula**, **P. Saha**, **S. Sarkar**, **P. Goyal**, and **A. Mukherjee** (2020). Hate begets hate: A temporal study of hate speech. *Proc. ACM Hum.-Comput. Interact.*, **4**(CSCW2). URL https://doi.org/10.1145/3415163.

238. **Mathew, B.**, **P. Saha**, **S. M. Yimam**, **C. Biemann**, **P. Goyal**, and **A. Mukherjee** (2021). Hatexplain: A benchmark dataset for explainable hate speech detection. *Proceedings*

of the AAAI Conference on Artificial Intelligence, **35**(17), 14867–14875. URL https://ojs.aaai.org/index.php/AAAI/article/view/17745.

239. **Menczer, F.**, **D. Crandall**, **Y.-Y. Ahn**, and **A. Kapadia** (2023). Addressing the harms of ai-generated inauthentic content. *Nature Machine Intelligence*, **5**(7), 679–680. ISSN 2522-5839. URL https://doi.org/10.1038/s42256-023-00690-w.

240. **Mendelsohn, J.** and **C. Budak**, When people are floods: Analyzing dehumanizing metaphors in immigration discourse with large language models. *In Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, Vienna, Austria, 2025. ISBN 979-8-89176-251-0. URL https://aclanthology.org/2025.acl-long.398/.

241. **Mendelsohn, J.**, **R. Le Bras**, **Y. Choi**, and **M. Sap**, From dogwhistles to bullhorns: Unveiling coded rhetoric with language models. *In Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, Toronto, Canada, 2023. URL https://aclanthology.org/2023.acl-long.845/.

242. **Merlo, L. I.**, **B. Chulvi**, **R. Ortega-Bueno**, and **P. Rosso** (2023). When humour hurts: linguistic features to foster explainability. *Procesamiento del Lenguaje Natural*, **70**(0), 85–98. ISSN 1989-7553. URL http://journal.sepln.org/sepln/ojs/ojs/index.php/pln/article/view/6481.

243. **Miller, G. A.** (1995). Wordnet: a lexical database for english. *Communications of the ACM*, **38**(11), 39–41. ISSN 1557-7317. URL http://dx.doi.org/10.1145/219717.219748.

244. **Min, S.**, **X. Lyu**, **A. Holtzman**, **M. Artetxe**, **M. Lewis**, **H. Hajishirzi**, and **L. Zettlemoyer**, Rethinking the role of demonstrations: What makes in-context learning work? *In Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Abu Dhabi, United Arab Emirates, 2022. URL https://aclanthology.org/2022.emnlp-main.759.

245. **Mohammad, S.**, **S. Kiritchenko**, **P. Sobhani**, **X. Zhu**, and **C. Cherry**, SemEval-2016 task 6: Detecting stance in tweets. *In Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*. Association for Computational Linguistics, San Diego, California, 2016. URL https://aclanthology.org/S16-1003.

246. **Mollas, I.**, **Z. Chrysopoulou**, **S. Karlos**, and **G. Tsoumakas** (2022). Ethos: a multi-label hate speech detection dataset. *Complex & Intelligent Systems*, **8**(6), 4663–4678. ISSN 2198-6053. URL https://doi.org/10.1007/s40747-021-00608-2.

247. **Morris, M. R.** (2024). Prompting considered harmful. *Commun. ACM*, **67**(12), 28–30. ISSN 0001-0782. URL https://doi.org/10.1145/3673861.

248. **Mozafari, M.**, **R. Farahbakhsh**, and **N. Crespi**, A bert-based transfer learning approach for hate speech detection in online social media. *In Complex Networks and Their Applications VIII*. Springer International Publishing, Cham, 2020*a*. ISBN 978-3-030-36687-2.

249. **Mozafari, M.**, **R. Farahbakhsh**, and **N. Crespi** (2020*b*). Hate speech detection and racial bias mitigation in social media based on bert model. *PloS one*, **15**(8), e0237861.

250. **Muhlmeyer, M.**, **J. Huang**, and **S. Agarwal** (2019). Event triggered social media chatter: A new modeling framework. *IEEE TCSS*, **6**(2), 197–207.

251. **Mukherjee, A.** and **H. Chang** (2023). The creative frontier of generative ai: Managing the novelty-usefulness tradeoff. *arXiv preprint arXiv:2306.03601*.

252. **Mulki, H.**, **H. Haddad**, **C. Bechikh Ali**, and **H. Alshabani**, L-HSAB: A Levantine Twitter dataset for hate speech and abusive language. *In Proceedings of the Third Workshop on Abusive Language Online*. Association for Computational Linguistics, Florence, Italy, 2019. URL https://aclanthology.org/W19-3512.

253. **Müller-Eberstein, M.**, **R. van der Goot**, and **B. Plank**, Probing for labeled dependency trees. *In Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, Dublin, Ireland, 2022. URL https://aclanthology.org/2022.acl-long.532/.

254. **Mun, J.**, **E. Allaway**, **A. Yerukola**, **L. Vianna**, **S.-J. Leslie**, and **M. Sap**, Beyond denouncing hate: Strategies for countering implied biases and stereotypes in language. *In Findings of the Association for Computational Linguistics: EMNLP 2023*. Association for Computational Linguistics, Singapore, 2023. URL https://aclanthology.org/2023.findings-emnlp.653.

255. **Mun, J.**, **C. Buerger**, **J. T. Liang**, **J. Garland**, and **M. Sap**, Counterspeakers' perspectives: Unveiling barriers and ai needs in the fight against online hate. *In Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems*, CHI '24. Association for Computing Machinery, New York, NY, USA, 2024. ISBN 9798400703300. URL https://doi.org/10.1145/3613904.3642025.

256. **Munn, L.** (2020). Angry by design: toxic communication and technical architectures. *Humanities and Social Sciences Communications*, **7**(1). URL https://doi.org/10.1057/s41599-020-00550-7.

257. **Myers, S. A.**, **C. Zhu**, and **J. Leskovec**, Information diffusion and external influence in networks. *In ACM SIGKDD*. 2012.

258. **Nadal, K. L.**, **R. King**, **D. G. Sissoko**, **N. Floyd**, and **D. Hines** (2021). The legacies of systemic and internalized oppression: Experiences of microaggressions, imposter phenomenon, and stereotype threat on historically marginalized groups. *New Ideas in Psychology*, **63**, 100895. ISSN 0732-118X. URL https://www.sciencedirect.com/science/article/pii/S0732118X21000441.

259. **Nadeem, M.**, **A. Bethke**, and **S. Reddy**, StereoSet: Measuring stereotypical bias in pretrained language 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)*. Association for Computational Linguistics, Online, 2021. URL https://aclanthology.org/2021.acl-long.416.

260. **Nangia, N.**, **C. Vania**, **R. Bhalerao**, and **S. R. Bowman**, CrowS-pairs: A challenge dataset for measuring social biases in masked language models. *In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, Online, 2020. URL https://aclanthology.org/2020.emnlp-main.154.

261. **Naous, T.**, **M. J. Ryan**, **A. Ritter**, and **W. Xu**, Having beer after prayer? measuring cultural bias in large language models. *In Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, Bangkok, Thailand, 2024. URL https://aclanthology.org/2024.acl-long.862/.

262. **Navarro, J. I.** (2013). The psychology of hatred. *The Open Criminology Journal*, **6**(1), 10–17. ISSN 1874-9178. URL http://dx.doi.org/10.2174/1874917801306010010.

263. **Nellis, G.** (2023). Election cycles and global religious intolerance. *Proceedings of the National Academy of Sciences*, **120**(1), e2213198120. URL https://www.pnas.org/doi/abs/10.1073/pnas.2213198120.

264. **Névéol, A.**, **Y. Dupont**, **J. Bezançon**, and **K. Fort**, French CrowS-pairs: Extending a challenge dataset for measuring social bias in masked language models to a language other than English. *In Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, Dublin, Ireland, 2022. URL https://aclanthology.org/2022.acl-long.583.

265. **Nguyen, D. Q.**, **T. Vu**, and **A. Tuan Nguyen**, BERTweet: A pre-trained language model for English tweets. *In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*. Association for Computational Linguistics, Online, 2020. URL https://aclanthology.org/2020.emnlp-demos.2.

266. **Nielsen, F.** (2019). On the jensen–shannon symmetrization of distances relying on abstract means. *Entropy*, **21**(5). ISSN 1099-4300. URL https://www.mdpi.com/1099-4300/21/5/485.

267. **Nogueira dos Santos, C.**, **I. Melnyk**, and **I. Padhi**, Fighting offensive language on social media with unsupervised text style transfer. *In ACL*. 2018.

268. **Noorian, Z.**, **A. Ghenai**, **H. Moradisani**, **F. Zarrinkalam**, and **S. Z. Alavijeh** (2024). User-centric modeling of online hate through the lens of psycholinguistic patterns and behaviors in social media. *IEEE Transactions on Computational Social Systems*, **11**(3), 4354–4366.

269. **Nozza, D.**, Exposing the limits of zero-shot cross-lingual hate speech detection. *In Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*. Association for Computational Linguistics, Online, 2021. URL https://aclanthology.org/2021.acl-short.114.

270. **Nozza, D.**, **C. Volpetti**, and **E. Fersini**, Unintended bias in misogyny detection. *In IEEE/WIC/ACM International Conference on Web Intelligence*. 2019.

271. **Ocampo, N. B.**, **E. Cabrio**, and **S. Villata**, Playing the part of the sharp bully: Generating adversarial examples for implicit hate speech detection. *In Findings of the Association for Computational Linguistics: ACL 2023*. Association for Computational Linguistics, Toronto, Canada, 2023*a*. URL https://aclanthology.org/2023.findings-acl.173.

272. **Ocampo, N. B.**, **E. Cabrio**, and **S. Villata**, Unmasking the hidden meaning: Bridging implicit and explicit hate speech embedding representations. *In Findings of the Association for Computational Linguistics: EMNLP 2023*. Association for Computational Linguistics, Singapore, 2023*b*. URL https://aclanthology.org/2023.findings-emnlp.441.

273. **Ocampo, N. B.**, **E. Sviridova**, **E. Cabrio**, and **S. Villata**, An in-depth analysis of implicit and subtle hate speech messages. *In Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*. Association for Computational Linguistics, Dubrovnik, Croatia, 2023*c*. URL https://aclanthology.org/2023.eacl-main.147.

274. **Oriol, B.**, **C. Canton-Ferrer**, and **X. G. i Nieto**, Hate speech in pixels: Detection of offensive memes towards automatic moderation. *In NeurIPS 2019 Workshop on AI for Social Good*. Vancouver, Canada, 2019.

275. **Orlikowski, M.**, **P. Röttger**, **P. Cimiano**, and **D. Hovy**, The ecological fallacy in annotation: Modeling human label variation goes beyond sociodemographics. *In Proc. of the 61st Annual Meeting of the ACL (Volume 2: Short Papers)*. ACL, Toronto, Canada, 2023. URL https://aclanthology.org/2023.acl-short.88.

276. **Ostyakova, L.**, **V. Smilga**, **K. Petukhova**, **M. Molchanova**, and **D. Kornev**, ChatGPT vs. crowdsourcing vs. experts: Annotating open-domain conversations with speech functions. *In Proc. of the 24th Meeting of the Special Interest Group on Discourse and Dialogue*. ACL, Prague, Czechia, 2023. URL https://aclanthology.org/2023.sigdial-1.23.

277. **Ousidhoum, N.**, **Z. Lin**, **H. Zhang**, **Y. Song**, and **D.-Y. Yeung**, Multilingual and multi-aspect hate speech analysis. *In Proc. of EMNLP*. ACL, 2019.

278. **Ousidhoum, N.**, **Y. Song**, and **D.-Y. Yeung**, Comparative evaluation of label agnostic selection bias in multilingual hate speech datasets. *In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. 2020.

279. **Ousidhoum, N.**, **X. Zhao**, **T. Fang**, **Y. Song**, and **D.-Y. Yeung**, Probing toxic content in large pre-trained language 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)*. Association for Computational Linguistics, Online, 2021. URL https://aclanthology.org/2021.acl-long.329.

280. **Ouyang, L.**, **J. Wu**, **X. Jiang**, **D. Almeida**, **C. Wainwright**, **P. Mishkin**, **C. Zhang**, **S. Agarwal**, **K. Slama**, **A. Ray**, **J. Schulman**, **J. Hilton**, **F. Kelton**, **L. Miller**, **M. Simens**, **A. Askell**, **P. Welinder**, **P. F. Christiano**, **J. Leike**, and **R. Lowe**, Training language models to follow instructions with human feedback. *In Advances in Neural Information Processing Systems*, volume 35. Curran Associates, Inc., 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/b1efde53be364a73914f58805a001731-Paper-Conference.pdf.

281. **Pan, S.**, **L. Luo**, **Y. Wang**, **C. Chen**, **J. Wang**, and **X. Wu** (2024). Unifying large language models and knowledge graphs: A roadmap. *IEEE Transactions on Knowledge and Data Engineering*, 1–20.

282. **Pan, Z.**, **D. Mehta**, **A. Tiwari**, **S. Ireddy**, **Z. Yang**, and **F. Jin**, An interactive platform to track global covid-19 epidemic. *In IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining, ASONAM 2020, The Hague, Netherlands, December 7-10, 2020*. IEEE, 2020. ISBN 978-1-7281-1056-1. URL https://doi.org/10.1109/ASONAM49781.2020.9381436.

283. **Papineni, K.**, **S. Roukos**, **T. Ward**, and **W.-J. Zhu**, Bleu: a method for automatic evaluation of machine translation. *In Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, Philadelphia, Pennsylvania, USA, 2002. URL https://aclanthology.org/P02-1040.

284. **Park, J. H.**, **J. Shin**, and **P. Fung**, Reducing gender bias in abusive language detection. *In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Brussels, Belgium, 2018. URL https://www.aclweb.org/anthology/D18-1302.

285. **Parker, S.** and **D. Ruths** (2023). Is hate speech detection the solution the world wants? *Proceedings of the National Academy of Sciences*, **120**(10), e2209384120. URL https://www.pnas.org/doi/abs/10.1073/pnas.2209384120.

286. **Pavlopoulos, J.**, **P. Malakasiotis**, and **I. Androutsopoulos**, Deep learning for user comment moderation. *In Proceedings of the First Workshop on Abusive Language Online*. Association for Computational Linguistics, Vancouver, BC, Canada, 2017. URL https://aclanthology.org/W17-3004.

287. **Pavlopoulos, J.**, **J. Sorensen**, **L. Laugier**, and **I. Androutsopoulos**, SemEval-2021 task 5: Toxic spans detection. *In Proceedings of the 15th International Workshop on Semantic Evaluation (SemEval-2021)*. Association for Computational Linguistics, Online, 2021. URL https://aclanthology.org/2021.semeval-1.6.

288. **Pawar, S.**, **J. Park**, **J. Jin**, **A. Arora**, **J. Myung**, **S. Yadav**, **F. G. Haznitrama**, **I. Song**, **A. Oh**, and **I. Augenstein** (2024). Survey of cultural awareness in language models: Text and beyond. URL https://arxiv.org/abs/2411.00860.

289. **Pennington, J.**, **R. Socher**, and **C. Manning**, GloVe: Global vectors for word representation. *In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, Doha, Qatar, 2014. URL https://aclanthology.org/D14-1162.

290. **Petkevic, V.** and **A. Nai** (2022). Political attacks in 280 characters or less: A new tool for the automated classification of campaign negativity on social media. *American Politics Research*, **50**(3), 279–302.

291. **Piazza, J. A.** (2020). Politician hate speech and domestic terrorism. *International Interactions*, **46**(3), 431–453.

292. **Piot, P.**, **P. Martín-Rodilla**, and **J. Parapar** (2024). Metahate: A dataset for unifying efforts on hate speech detection. *Proceedings of the International AAAI Conference on Web and Social Media*, **18**(1), 2025–2039. URL https://ojs.aaai.org/index.php/ICWSM/article/view/31445.

293. **Plaza-del Arco, F. M.**, **D. Nozza**, and **D. Hovy**, Respectful or toxic? using zero-shot learning with language models to detect hate speech. *In The 7th Workshop on Online Abuse and Harms (WOAH)*. Association for Computational Linguistics, Toronto, Canada, 2023. URL https://aclanthology.org/2023.woah-1.6.

294. **Poletto, F.**, **V. Basile**, **M. Sanguinetti**, **C. Bosco**, and **V. Patti** (2021). Resources and benchmark corpora for hate speech detection: a systematic review. *Language Resources and Evaluation*, **55**(2), 477–523. ISSN 1574-0218. URL https://doi.org/10.1007/s10579-020-09502-8.

295. **Polignano, M.**, **G. Colavito**, **C. Musto**, **M. de Gemmis**, and **G. Semeraro**, Lexicon enriched hybrid hate speech detection with human-centered explanations. *In Adjunct Proceedings of the 30th ACM Conference on User Modeling, Adaptation and Personalization*, UMAP '22 Adjunct. Association for Computing Machinery, New York, NY, USA, 2022. ISBN 9781450392327. URL https://doi.org/10.1145/3511047.3537688.

296. **Pramanick, S.**, **D. Dimitrov**, **R. Mukherjee**, **S. Sharma**, **M. S. Akhtar**, **P. Nakov**, and **T. Chakraborty**, Detecting harmful memes and their targets. *In Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*. Association for Computational Linguistics, Online, 2021. URL https://aclanthology.org/2021.findings-acl.246.

297. **Preoţiuc-Pietro, D.** and **L. Ungar**, User-level race and ethnicity predictors from twitter text. *In Proceedings of the 27th International Conference on Computational Linguistics*. 2018.

298. **Pretus, C.**, **J. L. Ray**, **Y. Granot**, **W. A. Cunningham**, and **J. J. Van Bavel** (2023). The psychology of hate: Moral concerns differentiate hate from dislike. *European Journal of Social Psychology*, **53**(2), 336–353. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/ejsp.2906.

299. **Pryzant, R.**, **R. Diehl Martinez**, **N. Dass**, **S. Kurohashi**, **D. Jurafsky**, and **D. Yang** (2020). Automatically neutralizing subjective bias in text. *AAAI*, 480–489. URL https://ojs.aaai.org/index.php/AAAI/article/view/5385.

300. **Qian, J.**, **A. Bethke**, **Y. Liu**, **E. Belding**, and **W. Y. Wang**, A benchmark dataset for learning to intervene in online hate speech. *In EMNLP-IJCNLP*. 2019.

301. **Qian, J.**, **M. ElSherief**, **E. Belding**, and **W. Y. Wang**, Leveraging intra-user and inter-user representation learning for automated hate speech detection. *In Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*. Association for Computational Linguistics, New Orleans, Louisiana, 2018. URL https://aclanthology.org/N18-2019.

302. **Qian, J.**, **H. Wang**, **M. ElSherief**, and **X. Yan**, Lifelong learning of hate speech classification on social media. *In Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, Online, 2021. URL https://aclanthology.org/2021.naacl-main.183.

303. **Rachmad, Y. E.** (2024). Social media impact theory. URL osf.io/2dbnx.

304. **Ramapuram, J.**, **M. Gregorova**, and **A. Kalousis** (2020). Lifelong generative modeling. *Neurocomputing*, **404**, 381–400. ISSN 0925-2312. URL https://www.sciencedirect.com/science/article/pii/S0925231220303623.

305. **Ranasinghe, T.** and **M. Zampieri** (2021). Multilingual offensive language identification for low-resource languages. *ACM Trans. Asian Low-Resour. Lang. Inf. Process.*, **21**(1). ISSN 2375-4699. URL https://doi.org/10.1145/3457610.

306. **Razo, D.** and **S. Kübler**, Investigating sampling bias in abusive language detection. *In Proceedings of the Fourth Workshop on Online Abuse and Harms*. 2020.

307. **Reid, M.** and **V. Zhong**, LEWIS: Levenshtein editing for unsupervised text style transfer. *In Findings of ACL-IJCNLP 2021*. 2021.

308. **Reimers, N.** and **I. Gurevych**, Sentence-BERT: Sentence embeddings using Siamese BERT-networks. *In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Association for Computational Linguistics, Hong Kong, China, 2019. URL https://aclanthology.org/D19-1410.

309. **Reyero Lobo, P.**, **E. Daga**, **H. Alani**, and **M. Fernandez**, *Knowledge-Grounded Target Group Language Recognition in Hate Speech*. IOS Press, 2023. ISBN 9781643684253. URL http://dx.doi.org/10.3233/SSW230002.

310. **Rheault, L.**, **E. Rayment**, and **A. Musulan** (2019). Politicians in the line of fire: Incivility and the treatment of women on social media. *Research & Politics*, **6**(1), 2053168018816228.

311. **Ribeiro, M. H.**, **P. H. Calais**, **Y. A. Santos**, **V. A. F. Almeida**, and **W. M. Jr.**, Characterizing and detecting hateful users on twitter. *In ICWSM*. 2018.

312. **Ribeiro, M. H.**, **R. West**, **R. Lewis**, and **S. Kairam** (2025). Post guidance for online communities. URL https://arxiv.org/abs/2411.16814.

313. **Ribeiro, M. T.**, **S. Singh**, and **C. Guestrin**, "Why Should I Trust You?": Explaining the Predictions of Any Classifier. *In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16. Association for Computing Machinery, New York, NY, USA, 2016. ISBN 9781450342322. URL https://doi.org/10.1145/2939672.2939778.

314. **Richie, R.**, **S. Grover**, and **F. R. Tsui**, Inter-annotator agreement is not the ceiling of machine learning performance: Evidence from a comprehensive set of simulations. *In Proceedings of the 21st Workshop on Biomedical Language Processing*. Association for Computational Linguistics, Dublin, Ireland, 2022. URL https://aclanthology.org/2022.bionlp-1.26.

315. **Rippel, O.**, **M. Paluri**, **P. Dollár**, and **L. D. Bourdev**, Metric learning with adaptive density discrimination. *In 4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*. 2016. URL http://arxiv.org/abs/1511.05939.

316. **Rizwan, H.**, **M. H. Shakeel**, and **A. Karim**, Hate-speech and offensive language detection in Roman Urdu. *In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, Online, 2020. URL https://aclanthology.org/2020.emnlp-main.197.

317. **Robertson, S.** and **H. Zaragoza** (2009). The probabilistic relevance framework: Bm25 and beyond. *Found. Trends Inf. Retr.*, **3**(4), 333–389. ISSN 1554-0669. URL https://doi.org/10.1561/1500000019.

318. **Romim, N.**, **M. Ahmed**, **H. Talukder**, and **M. S. Islam**, Hate speech detection in the bengali language: A dataset and its baseline evaluation. *In Algorithms for Intelligent Systems*. Springer Singapore, 2021, 457–468. URL https://doi.org/10.1007/978-981-16-0586-4_37.

319. **Ross, B.**, **M. Rist**, **G. Carbonell**, **B. Cabrera**, **N. Kurowsky**, and **M. Wojatzki**, Measuring the reliability of hate speech annotations: The case of the european refugee crisis. *In NLP4CMC III: 3rd Workshop on Natural Language Processing for Computer-Mediated Communication*, Bochumer Linguistische Arbeitsberichte. Ruhr-Universitat Bochum, Germany, 2016. URL https://sites.google.com/site/nlp4cmc2016/.

320. **Rossetti, G.**, **L. Milli**, **S. Rinzivillo**, **A. Sîrbu**, **D. Pedreschi**, and **F. Giannotti** (2018). Ndlib: a python library to model and analyze diffusion processes over complex networks. *International Journal of Data Science and Analytics*, **5**(1), 61–79.

321. **Röttger, P.**, **H. Seelawi**, **D. Nozza**, **Z. Talat**, and **B. Vidgen**, Multilingual HateCheck: Functional tests for multilingual hate speech detection models. *In Proceedings of the Sixth Workshop on Online Abuse and Harms (WOAH)*. Association for Computational Linguistics, Seattle, Washington (Hybrid), 2022. URL https://aclanthology.org/2022.woah-1.15/.

322. **Rottger, P.**, **B. Vidgen**, **D. Hovy**, and **J. Pierrehumbert**, Two contrasting data annotation paradigms for subjective NLP tasks. *In Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, Seattle, United States, 2022. URL https://aclanthology.org/2022.naacl-main.13.

323. **Röttger, P.**, **B. Vidgen**, **D. Nguyen**, **Z. Waseem**, **H. Margetts**, and **J. Pierrehumbert**, 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)*. Association for Computational Linguistics, Online, 2021. URL https://aclanthology.org/2021.acl-long.4.

324. **Roy, S.**, **A. Harshvardhan**, **A. Mukherjee**, and **P. Saha**, Probing LLMs for hate speech detection: strengths and vulnerabilities. *In Findings of the Association for Computational Linguistics: EMNLP 2023*. Association for Computational Linguistics, Singapore, 2023. URL https://aclanthology.org/2023.findings-emnlp.407.

325. **Roychowdhury, S.** and **V. Gupta**, Data-efficient methods for improving hate speech detection. *In Findings of the Association for Computational Linguistics: EACL 2023*. Association for Computational Linguistics, Dubrovnik, Croatia, 2023. URL https://aclanthology.org/2023.findings-eacl.9/.

326. **Rozemberczki, B.** and **R. Sarkar**, Characteristic functions on graphs: Birds of a feather, from statistical descriptors to parametric models. *In Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, CIKM '20. Association for Computing Machinery, New York, NY, USA, 2020. ISBN 9781450368599. URL https://doi.org/10.1145/3340531.3411866.

327. **Sachdeva, P.**, **R. Barreto**, **G. Bacon**, **A. Sahn**, **C. von Vacano**, and **C. Kennedy**, The measuring hate speech corpus: Leveraging rasch measurement theory for data perspectivism. *In Proceedings of the 1st Workshop on Perspectivist Approaches to NLP @LREC2022*. European Language Resources Association, Marseille, France, 2022. URL https://aclanthology.org/2022.nlperspectives-1.11.

328. **Saha, K.**, **E. Chandrasekharan**, and **M. De Choudhury** (2019). Prevalence and psychological effects of hateful speech in online college communities. *Proceedings of the ACM Web Science Conference. ACM Web Science Conference*, **2019**, 255–264. URL https://doi.org/10.1145/3292522.3326032.

329. **Sahnan, D.**, **S. Dahiya**, **V. Goel**, **A. Bandhakavi**, and **T. Chakraborty**, Better prevent than react: Deep stratified learning to predict hate intensity of twitter reply chains. *In 2021 IEEE International Conference on Data Mining (ICDM)*. 2021.

330. **Sahnan, D.**, **V. Goel**, **S. Masud**, **C. Jain**, **V. Goyal**, and **T. Chakraborty** (2022). Diva: A scalable, interactive and customizable visual analytics platform for information diffusion on large networks. *ACM Trans. Knowl. Discov. Data*. ISSN 1556-4681. URL https://doi.org/10.1145/3558771. Just Accepted.

331. **Sahoo, N.**, **P. Kulkarni**, **A. Ahmad**, **T. Goyal**, **N. Asad**, **A. Garimella**, and **P. Bhattacharyya**, IndiBias: A benchmark dataset to measure social biases in language models for Indian context. *In Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*. Association for Computational Linguistics, Mexico City, Mexico, 2024. URL https://aclanthology.org/2024.naacl-long.487/.

332. **Saikh, T.**, **S. Barman**, **H. Kumar**, **S. Sahu**, and **S. Palit**, Emojis trash or treasure: Utilizing emoji to aid hate speech detection. *In Proceedings of the 21st International Conference on Natural Language Processing (ICON)*. NLP Association of India (NLPAI), AU-KBC Research Centre, Chennai, India, 2024. URL https://aclanthology.org/2024.icon-1.64/.

333. **Salminen, J.**, **H. Almerekhi**, **M. Milenković**, **S.-g. Jung**, **J. An**, **H. Kwak**, and **B. Jansen** (2018). Anatomy of online hate: Developing a taxonomy and machine learning models for identifying and classifying hate in online news media. *Proceedings of the International AAAI Conference on Web and Social Media*, **12**(1). URL https://ojs.aaai.org/index.php/ICWSM/article/view/15028.

334. **Sap, M.**, **R. L. Bras**, **E. Allaway**, **C. Bhagavatula**, **N. Lourie**, **H. Rashkin**, **B. Roof**, **N. A. Smith**, and **Y. Choi** (2018). ATOMIC: an atlas of machine commonsense for if-then reasoning. *CoRR*, **abs/1811.00146**. URL http://arxiv.org/abs/1811.00146.

335. **Sap, M.**, **D. Card**, **S. Gabriel**, **Y. Choi**, and **N. A. Smith**, The risk of racial bias in hate speech detection. *In Proc. of the 57th Annual Meeting of the ACL*. ACL, Florence, Italy, 2019. URL https://aclanthology.org/P19-1163.

336. **Sap, M.**, **S. Gabriel**, **L. Qin**, **D. Jurafsky**, **N. A. Smith**, and **Y. Choi**, Social bias frames: Reasoning about social and power implications of language. *In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, Online, 2020. URL https://aclanthology.org/2020.acl-main.486.

337. **Sap, M.**, **S. Swayamdipta**, **L. Vianna**, **X. Zhou**, **Y. Choi**, and **N. A. Smith**, Annotators with attitudes: How annotator beliefs and identities bias toxic language detection. *In Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, Seattle, United States, 2022. URL https://aclanthology.org/2022.naacl-main.431.

338. **Sarwar, S. M.** and **V. Murdock** (2022). Unsupervised domain adaptation for hate speech detection using a data augmentation approach. *Proceedings of the International AAAI Conference on Web and Social Media*, **16**(1), 852–862. URL https://ojs.aaai.org/index.php/ICWSM/article/view/19340.

339. **Schäfer, J.** and **E. Kistner**, HS-EMO: Analyzing emotions in hate speech. *In Proceedings of the 19th Conference on Natural Language Processing (KONVENS 2023)*. Association for Computational Linguistics, Ingolstadt, Germany, 2023. URL https://aclanthology.org/2023.konvens-main.17/.

340. **Scheuerman, M. K.**, **J. A. Jiang**, **C. Fiesler**, and **J. R. Brubaker** (2021). A framework of severity for harmful content online. *Proc. ACM Hum.-Comput. Interact.*, **5**(CSCW2). URL https://doi.org/10.1145/3479512.

341. **Schmaltz, A.** (2021). Detecting local insights from global labels: Supervised and zero-shot sequence labeling via a convolutional decomposition. *Computational Linguistics*, **47**(4), 729–773. URL https://aclanthology.org/2021.cl-4.25.

342. **Schmidhuber, J.** (1997). Long short-term memory. *Neural Computation*, **9**(8), 1735–1780.

343. **Schmidt, A.** and **M. Wiegand**, A survey on hate speech detection using natural language processing. *In Proceedings of the Fifth International Workshop on Natural Language Processing for Social Media*. Association for Computational Linguistics, Valencia, Spain, 2017. URL https://aclanthology.org/W17-1101.

344. **Schneider, P.**, **T. Schopf**, **J. Vladika**, **M. Galkin**, **E. Simperl**, and **F. Matthes**, A decade of knowledge graphs in natural language processing: A survey. *In Proceedings of the 2nd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 12th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. Association for Computational Linguistics, Online only, 2022. URL https://aclanthology.org/2022.aacl-main.46.

345. **Schroff, F.**, **D. Kalenichenko**, and **J. Philbin**, Facenet: A unified embedding for face recognition and clustering. *In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2015.

346. **Schuster, T.**, **D. Shah**, **Y. J. S. Yeo**, **D. Roberto Filizzola Ortiz**, **E. Santus**, and **R. Barzilay**, Towards debiasing fact verification models. *In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Association for Computational Linguistics, Hong Kong, China, 2019. URL https://www.aclweb.org/anthology/D19-1341.

347. **Sellam, T.**, **S. Yadlowsky**, **I. Tenney**, **J. Wei**, **N. Saphra**, **A. D'Amour**, **T. Linzen**, **J. Bastings**, **I. R. Turc**, **J. Eisenstein**, **D. Das**, and **E. Pavlick**, The multiBERTs: BERT

reproductions for robustness analysis. *In International Conference on Learning Representations*. 2022. URL https://openreview.net/forum?id=K0E_F0gFDgA.

348. **Shah, D. S.**, **H. A. Schwartz**, and **D. Hovy**, Predictive biases in natural language processing models: A conceptual framework and overview. *In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, Online, 2020. URL https://aclanthology.org/2020.acl-main.468.

349. **Sharma, S.**, **F. Alam**, **M. S. Akhtar**, **D. Dimitrov**, **G. Da San Martino**, **H. Firooz**, **A. Halevy**, **F. Silvestri**, **P. Nakov**, and **T. Chakraborty**, Detecting and understanding harmful memes: A survey. *In Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI-22*. International Joint Conferences on Artificial Intelligence Organization, 2022. URL https://doi.org/10.24963/ijcai.2022/781. Survey Track.

350. **Shen, T.**, **T. Lei**, **R. Barzilay**, and **T. Jaakkola**, Style transfer from non-parallel text by cross-alignment. *In NeurIPS*, volume 30. Curran Associates, Inc., 2017.

351. **Siddarth, D.**, **R. Shankar**, and **J. Pal** (2021). 'we do politics so we can change politics': communication strategies and practices in the aam aadmi party's institutionalization process. *Information, Communication & Society*, **24**(10), 1361–1381. URL https://doi.org/10.1080/1369118X.2020.1856910.

352. **Singhal, K.**, **S. Azizi**, **T. Tu**, **S. S. Mahdavi**, **J. Wei**, **H. W. Chung**, **N. Scales**, **A. Tanwani**, **H. Cole-Lewis**, **S. Pfohl**, **P. Payne**, **M. Seneviratne**, **P. Gamble**, **C. Kelly**, **N. Scharli**, **A. Chowdhery**, **P. Mansfield**, **B. A. y Arcas**, **D. Webster**, **G. S. Corrado**, **Y. Matias**, **K. Chou**, **J. Gottweis**, **N. Tomasev**, **Y. Liu**, **A. Rajkomar**, **J. Barral**, **C. Semturs**, **A. Karthikesalingam**, and **V. Natarajan** (2022). Large language models encode clinical knowledge.

353. **Slovic, P.**, **B. Fischhoff**, and **S. Lichtenstein** (1977). Behavioral decision theory. *Annual Review of Psychology*, **28**(Volume 28, 1977), 1–39. ISSN 1545-2085. URL https://www.annualreviews.org/content/journals/10.1146/annurev.ps.28.020177.000245.

354. **Solovev, K.** and **N. Pröllochs**, Hate speech in the political discourse on social media: Disparities across parties, gender, and ethnicity. *In Proceedings of the ACM Web Conference 2022*, WWW '22. Association for Computing Machinery, New York, NY, USA, 2022. ISBN 9781450390965. URL https://doi.org/10.1145/3485447.3512261.

355. **Speer, R.** and **C. Havasi**, Representing general relational knowledge in ConceptNet 5. *In Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*. European Language Resources Association (ELRA), Istanbul, Turkey, 2012. URL http://www.lrec-conf.org/proceedings/lrec2012/pdf/1072_Paper.pdf.

356. **Spence, R.**, **A. Bifulco**, **P. Bradbury**, **E. Martellozzo**, and **J. DeMarco** (2024). Content moderator mental health, secondary trauma, and well-being: A cross-sectional study. *Cyberpsychology, Behavior, and Social Networking*, **27**(2), 149–155. URL https://doi.org/10.1089/cyber.2023.0298. PMID: 38153846.

357. **Spertus, E.**, Smokey: automatic recognition of hostile messages. *In Proceedings of the Fourteenth National Conference on Artificial Intelligence and Ninth Conference on Innovative Applications of Artificial Intelligence*, AAAI'97/IAAI'97. AAAI Press, 1997. ISBN 0262510952.

358. **Sridhar, R.** and **D. Yang**, Explaining toxic text via knowledge enhanced text generation. *In Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, Seattle, United States, 2022. URL `https://aclanthology.org/2022.naacl-main.59`.

359. **Stamou, V.**, **I. Alexiou**, **A. Klimi**, **E. Molou**, **A. Saivanidou**, and **S. Markantonatou**, Cleansing & expanding the HURTLEX(el) with a multidimensional categorization of offensive words. *In Proceedings of the Sixth Workshop on Online Abuse and Harms (WOAH)*. Association for Computational Linguistics, Seattle, Washington (Hybrid), 2022. URL `https://aclanthology.org/2022.woah-1.10`.

360. **Steels, L.** (2016). Human language is a culturally evolving system. *Psychonomic Bulletin & Review*, **24**(1), 190–193. URL `https://doi.org/10.3758/s13423-016-1086-6`.

361. **Steiger, M.**, **T. J. Bharucha**, **S. Venkatagiri**, **M. J. Riedl**, and **M. Lease**, The psychological well-being of content moderators: The emotional labor of commercial moderation and avenues for improving support. *In Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, CHI '21. Association for Computing Machinery, New York, NY, USA, 2021. ISBN 9781450380966. URL `https://doi.org/10.1145/3411764.3445092`.

362. **Su, H.-P.**, **Z.-J. Huang**, **H.-T. Chang**, and **C.-J. Lin**, Rephrasing profanity in Chinese text. *In Proceedings of the First Workshop on Abusive Language Online*. Vancouver, BC, Canada, 2017.

363. **Suler, J.** (2004). The online disinhibition effect. *Cyberpsychology & behavior : the impact of the Internet, multimedia and virtual reality on behavior and society*, **7 3**, 321–326.

364. **Sun, C.**, **X. Qiu**, **Y. Xu**, and **X. Huang**, How to fine-tune bert for text classification? *In Chinese Computational Linguistics*. Springer International Publishing, Cham, 2019*a*. ISBN 978-3-030-32381-3.

365. **Sun, G.**, **T. Tang**, **T.-Q. Peng**, **R. Liang**, and **Y. Wu** (2017). Socialwave: Visual analysis of spatio-temporal diffusion of information on social media. *ACM Trans. Intell. Syst. Technol.*, **9**(2). ISSN 2157-6904. URL `https://doi.org/10.1145/3106775`.

366. **Sun, T.**, **A. Gaut**, **S. Tang**, **Y. Huang**, **M. ElSherief**, **J. Zhao**, **D. Mirza**, **E. Belding**, **K.-W. Chang**, and **W. Y. Wang**, Mitigating gender bias in natural language processing: Literature review. *In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, Florence, Italy, 2019*b*. URL `https://aclanthology.org/P19-1159`.

367. **Suresh, H.** and **J. Guttag**, A framework for understanding sources of harm throughout the machine learning life cycle. *In Proceedings of the 1st ACM Conference on Equity and Access in Algorithms, Mechanisms, and Optimization*, EAAMO '21. Association

for Computing Machinery, New York, NY, USA, 2021. ISBN 9781450385534. URL https://doi.org/10.1145/3465416.3483305.

368. **Sánchez-Junquera, J.**, **B. Chulvi**, **P. Rosso**, and **S. P. Ponzetto** (2021). How do you speak about immigrants? taxonomy and stereoimmigrants dataset for identifying stereotypes about immigrants. *Applied Sciences*, **11**(8). ISSN 2076-3417. URL https://www.mdpi.com/2076-3417/11/8/3610.

369. **Tahmasbi, N.** and **E. Rastegari** (2018). A socio-contextual approach in automated detection of public cyberbullying on twitter. *Trans. Soc. Comput.*, **1**(4). URL https://doi.org/10.1145/3290838.

370. **Tay, Y.**, **A. T. Luu**, **A. Zhang**, **S. Wang**, and **S. C. Hui**, Compositional de-attention networks. *In Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper_files/paper/2019/file/16fc18d787294ad5171100e33d05d4e2-Paper.pdf.

371. **Than, N.**, **M. Y. Rodriguez**, **D. Yoong**, and **F. Windel** (2020). Welcome to gab alt right discourses.

372. **Tonneau, M.**, **D. Liu**, **N. Malhotra**, **S. A. Hale**, **S. P. Fraiberger**, **V. Orozco-Olvera**, and **P. Röttger** (2024). Hateday: Insights from a global hate speech dataset representative of a day on twitter. URL https://arxiv.org/abs/2411.15462.

373. **Toraman, C.**, **F. Şahinuç**, and **E. H. Yılmaz** (2022). Large-scale hate speech detection with cross-domain transfer.

374. **Touvron, H.**, **T. Lavril**, **G. Izacard**, **X. Martinet**, **M.-A. Lachaux**, **T. Lacroix**, **B. Rozière**, **N. Goyal**, **E. Hambro**, **F. Azhar**, **A. Rodriguez**, **A. Joulin**, **E. Grave**, and **G. Lample** (2023). Llama: Open and efficient foundation language models.

375. **Tran, M.**, **Y. Zhang**, and **M. Soleymani**, Towards a friendly online community: An unsupervised style transfer framework for profanity redaction. *In Coling*. 2020.

376. **Tsesis, A.**, *Destructive Messages: How Hate Speech Paves the Way For Harmful Social Movements*. Number v. 778 in Critical America. NYU Press, 2002. ISBN 9780814782729. URL https://books.google.co.in/books?id=QNgVCgAAQBAJ.

377. **Tunstall, L.**, **E. Beeching**, **N. Lambert**, **N. Rajani**, **K. Rasul**, **Y. Belkada**, **S. Huang**, **L. von Werra**, **C. Fourrier**, **N. Habib**, *et al.* (2023). Zephyr: Direct distillation of lm alignment. *arXiv preprint arXiv:2310.16944*. URL https://arxiv.org/abs/2310.16944.

378. **Vaidya, A.**, **F. Mai**, and **Y. Ning** (2020). Empirical analysis of multi-task learning for reducing identity bias in toxic comment detection. *Proceedings of the International AAAI Conference on Web and Social Media*, **14**(1), 683–693. URL https://ojs.aaai.org/index.php/ICWSM/article/view/7334.

379. **van Aken, B.**, **B. Winter**, **A. Löser**, and **F. A. Gers**, How does bert answer questions? a layer-wise analysis of transformer representations. *In Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, CIKM '19. Association for Computing Machinery, New York, NY, USA, 2019. ISBN 9781450369763. URL https://doi.org/10.1145/3357384.3358028.

380. **Van Alstyne, M.** (2024). Free speech vs. free ride: Navigating the supreme court's social media paradox. *Commun. ACM*, **67**(11), 29–31. ISSN 0001-0782. URL https://doi.org/10.1145/3696194.

381. **Van Alstyne, M.**, **M. D. Smith**, and **H. Lin** (2023). Improving section 230, preserving democracy, and protecting free speech. *Commun. ACM*, **66**(4), 26–28. ISSN 0001-0782. URL https://doi.org/10.1145/3584710.

382. **Van Der Wal, O.**, **J. Jumelet**, **K. Schulz**, and **W. Zuidema**, The birth of bias: A case study on the evolution of gender bias in an English language model. *In Proceedings of the 4th Workshop on Gender Bias in Natural Language Processing (GeBNLP)*. Association for Computational Linguistics, Seattle, Washington, 2022. URL https://aclanthology.org/2022.gebnlp-1.8.

383. **Van Hee, C.**, **E. Lefever**, and **V. Hoste**, SemEval-2018 task 3: Irony detection in English tweets. *In Proceedings of the 12th International Workshop on Semantic Evaluation*. Association for Computational Linguistics, New Orleans, Louisiana, 2018. URL https://aclanthology.org/S18-1005.

384. **Vaswani, A.**, **N. Shazeer**, **N. Parmar**, **J. Uszkoreit**, **L. Jones**, **A. N. Gomez**, **L. u. Kaiser**, and **I. Polosukhin**, Attention is all you need. *In Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

385. **Vendrell Ferran, Í.** (2024). Hate: toward a four-types model. *Review of Philosophy and Psychology*, **15**(2), 441–459. ISSN 1878-5166. URL https://doi.org/10.1007/s13164-021-00568-z.

386. **Venturott, L. I.** and **P. M. Ciarelli**, Data augmentation for improving hate speech detection on social networks. *In Proceedings of the Brazilian Symposium on Multimedia and the Web*, WebMedia '20. Association for Computing Machinery, New York, NY, USA, 2020. ISBN 9781450381963. URL https://doi.org/10.1145/3428658.3431760.

387. **Vidgen, B.** and **L. Derczynski** (2021). Directions in abusive language training data, a systematic review: Garbage in, garbage out. *PLOS ONE*, **15**(12), 1–32. URL https://doi.org/10.1371/journal.pone.0243300.

388. **Vidgen, B.**, **D. Nguyen**, **H. Margetts**, **P. Rossini**, and **R. Tromble**, Introducing CAD: the contextual abuse dataset. *In Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, Online, 2021*a*. URL https://aclanthology.org/2021.naacl-main.182.

389. **Vidgen, B.**, **T. Thrush**, **Z. Waseem**, and **D. Kiela**, Learning from the worst: Dynamically generated datasets to improve online hate detection. *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)*. Association for Computational Linguistics, Online, 2021*b*. URL https://aclanthology.org/2021.acl-long.132.

390. **Vijayaraghavan, P.**, **H. Larochelle**, and **D. Roy** (2021). Interpretable multi-modal hate speech detection.

391. **Vo, C. N.**, **K. B. Huynh**, **S. T. Luu**, and **T.-H. Do** (2025). Vithsd: exploiting hatred by targets for hate speech detection on vietnamese social media texts. *Journal of Computational Social Science*, **8**(2), 30. ISSN 2432-2725. URL https://doi.org/10.1007/s42001-024-00348-6.

392. **Wang, A.**, **A. Singh**, **J. Michael**, **F. Hill**, **O. Levy**, and **S. R. Bowman**, GLUE: A multi-task benchmark and analysis platform for natural language understanding. *In International Conference on Learning Representations*. 2019. URL https://openreview.net/forum?id=rJ4km2R5t7.

393. **Wang, H.**, **T. R. Yang**, **U. Naseem**, and **R. K.-W. Lee**, Multihateclip: A multilingual benchmark dataset for hateful video detection on youtube and bilibili. *In Proceedings of the 32nd ACM International Conference on Multimedia*, MM '24. Association for Computing Machinery, New York, NY, USA, 2024. ISBN 9798400706868. URL https://doi.org/10.1145/3664647.3681521.

394. **Wang, J.**, **V. W. Zheng**, **Z. Liu**, and **K. C. Chang**, Topological recurrent neural network for diffusion prediction. *In ICDM*. 2017.

395. **Wang, Y.**, **S. Zannettou**, **J. Blackburn**, **B. Bradlyn**, **E. De Cristofaro**, and **G. Stringhini**, A multi-platform analysis of political news discussion and sharing on web communities. *In 2021 IEEE International Conference on Big Data (Big Data)*. 2021.

396. **Wang, Z.** and **W. Li**, Hierarchical diffusion attention network. *In IJCAI*. 2019.

397. **Waseem, Z.**, Are you a racist or am I seeing things? annotator influence on hate speech detection on Twitter. *In Proc. of the First Workshop on NLP and Computational Social Science*. ACL, Austin, Texas, 2016. URL https://aclanthology.org/W16-5618.

398. **Waseem, Z.**, **T. Davidson**, **D. Warmsley**, and **I. Weber**, Understanding abuse: A typology of abusive language detection subtasks. *In Proceedings of the First Workshop on Abusive Language Online*. 2017.

399. **Waseem, Z.** and **D. Hovy**, Hateful symbols or hateful people? predictive features for hate speech detection on Twitter. *In Proceedings of the NAACL Student Research Workshop*. Association for Computational Linguistics, San Diego, California, 2016. URL https://aclanthology.org/N16-2013.

400. **Weerasooriya, T.**, **S. Dutta**, **T. Ranasinghe**, **M. Zampieri**, **C. Homan**, and **A. KhudaBukhsh**, Vicarious offense and noise audit of offensive speech classifiers: Unifying human and machine disagreement on what is offensive. *In Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Singapore, 2023. URL https://aclanthology.org/2023.emnlp-main.713.

401. **Wei, J.**, **X. Wang**, **D. Schuurmans**, **M. Bosma**, **b. ichter**, **F. Xia**, **E. Chi**, **Q. V. Le**, and **D. Zhou**, Chain-of-thought prompting elicits reasoning in large language models. *In Advances in Neural Information Processing Systems*, volume 35. 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/9d5609613524ecf4f15af0f7b31abca4-Paper-Conference.pdf.

402. **Weidinger, L.**, **J. Mellor**, **M. Rauh**, **C. Griffin**, **J. Uesato**, **P.-S. Huang**, **M. Cheng**, **M. Glaese**, **B. Balle**, **A. Kasirzadeh**, **Z. Kenton**, **S. Brown**, **W. Hawkins**, **T. Stepleton**, **C. Biles**, **A. Birhane**, **J. Haas**, **L. Rimell**, **L. A. Hendricks**, **W. Isaac**, **S. Legassick**, **G. Irving**, and **I. Gabriel** (2021). Ethical and social risks of harm from language models.

403. **Wenzek, G.**, **M.-A. Lachaux**, **A. Conneau**, **V. Chaudhary**, **F. Guzmán**, **A. Joulin**, and **E. Grave**, CCNet: Extracting high quality monolingual datasets from web crawl data. *In Proceedings of the Twelfth Language Resources and Evaluation Conference*. European Language Resources Association, Marseille, France, 2020. ISBN 979-10-95546-34-4. URL `https://aclanthology.org/2020.lrec-1.494/`.

404. **Wich, M.**, **H. Al Kuwatly**, and **G. Groh**, Investigating annotator bias with a graph-based approach. *In Proceedings of the Fourth Workshop on Online Abuse and Harms*. 2020.

405. **Wich, M.**, **C. Widmer**, **G. Hagerer**, and **G. Groh**, Investigating annotator bias in abusive language datasets. *In Proc. of the International Conference on Recent Advances in Natural Language Processing (RANLP 2021)*. INCOMA Ltd., Held Online, 2021. URL `https://aclanthology.org/2021.ranlp-1.170`.

406. **Wiegand, M.**, **J. Ruppenhofer**, and **E. Eder**, 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*. Association for Computational Linguistics, Online, 2021. URL `https://aclanthology.org/2021.naacl-main.48`.

407. **Wiegand, M.**, **J. Ruppenhofer**, and **T. Kleinbauer**, Detection of abusive language: the problem of biased datasets. *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)*. 2019.

408. **Williams, M. L.**, **P. Burnap**, **A. Javed**, **H. Liu**, and **S. Ozalp** (2019). Hate in the Machine: Anti-Black and Anti-Muslim Social Media Posts as Predictors of Offline Racially and Religiously Aggravated Crime. *The British Journal of Criminology*, **60**(1), 93–117. ISSN 0007-0955. URL `https://doi.org/10.1093/bjc/azz049`.

409. **Wojatzki, M. M.**, Do women perceive hate differently: Examining the relationship between hate speech, gender, and agreement judgments. *In Proceedings of the 14th Conference on Natural Language Processing (KONVENS 2018)*. Deutsche Nationalbibliothek, 2018.

410. **Wolf, T.**, **L. Debut**, **V. Sanh**, **J. Chaumond**, **C. Delangue**, **A. Moi**, **P. Cistac**, **T. Rault**, **R. Louf**, **M. Funtowicz**, **J. Davison**, **S. Shleifer**, **P. von Platen**, **C. Ma**, **Y. Jernite**, **J. Plu**, **C. Xu**, **T. Le Scao**, **S. Gugger**, **M. Drame**, **Q. Lhoest**, and **A. Rush**, Transformers: State-of-the-art natural language processing. *In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*. Association for Computational Linguistics, Online, 2020. URL `https://aclanthology.org/2020.emnlp-demos.6`.

411. **Wong, F.** and **R. Halgin** (2006). The "model minority": Bane or blessing for asian americans? *Journal of Multicultural Counseling and Development*, **34**(1), 38–49. URL `https://doi.org/10.1002/j.2161-1912.2006.tb00025.x`.

412. **Wu, Y.**, **S. Liu**, **K. Yan**, **M. Liu**, and **F. Wu** (2014). Opinionflow: Visual analysis of opinion diffusion on social media. *IEEE Transactions on Visualization and Computer Graphics*, **20**(12), 1763–1772.

413. **Wulczyn, E.**, **N. Thain**, and **L. Dixon**, Ex machina: Personal attacks seen at scale. *In Proceedings of the 26th International Conference on World Wide Web*, WWW '17. International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, CHE, 2017. ISBN 9781450349130. URL https://doi.org/10.1145/3038912.3052591.

414. **Xia, M.**, **A. Field**, and **Y. Tsvetkov**, Demoting racial bias in hate speech detection. *In Proc. of the Eighth International Workshop on Natural Language Processing for Social Media*. ACL, Online, 2020. URL https://aclanthology.org/2020.socialnlp-1.2.

415. **Xu, J.**, **X. Sun**, **Q. Zeng**, **X. Zhang**, **X. Ren**, **H. Wang**, and **W. Li**, Unpaired sentiment-to-sentiment translation: A cycled reinforcement learning approach. *In ACL*. 2018.

416. **Yadav, N.**, **S. Masud**, **V. Goyal**, **M. S. Akhtar**, and **T. Chakraborty**, Tox-BART: Leveraging toxicity attributes for explanation generation of implicit hate speech. *In Findings of the Association for Computational Linguistics: ACL 2024*. Association for Computational Linguistics, Bangkok, Thailand, 2024. URL https://aclanthology.org/2024.findings-acl.831.

417. **Yang, C.**, **M. Sun**, **H. Liu**, **S. Han**, **Z. Liu**, and **H. Luan** (2018). Neural diffusion model for microscopic cascade prediction. *ArXiv*, **abs/1812.08933**.

418. **Yang, C.**, **J. Tang**, **M. Sun**, **G. Cui**, and **Z. Liu**, Multi-scale information diffusion prediction with reinforced recurrent networks. *In IJCAI*. 2019.

419. **Yang, W.**, **Y. Wei**, **H. Wei**, **Y. Chen**, **G. Huang**, **X. Li**, **R. Li**, **N. Yao**, **X. Wang**, **X. Gu**, **M. B. Amin**, and **B. Kang** (2023*a*). Survey on explainable ai: From approaches, limitations and applications aspects. *Human-Centric Intelligent Systems*, **3**(3), 161–188. ISSN 2667-1336. URL https://doi.org/10.1007/s44230-023-00038-y.

420. **Yang, Y.**, **J. Kim**, **Y. Kim**, **N. Ho**, **J. Thorne**, and **S.-Y. Yun**, HARE: Explainable hate speech detection with step-by-step reasoning. *In Findings of the Association for Computational Linguistics: EMNLP 2023*. Association for Computational Linguistics, Singapore, 2023*b*. URL https://aclanthology.org/2023.findings-emnlp.365.

421. **Yang, Z.**, **J. Xu**, **Z. Pan**, and **F. Jin**, Covid19 tracking: An interactive tracking, visualizing and analyzing platform. *In IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining, ASONAM 2020, The Hague, Netherlands, December 7-10, 2020*. IEEE, 2020. ISBN 978-1-7281-1056-1. URL https://doi.org/10.1109/ASONAM49781.2020.9381414.

422. **Yin, W.** and **A. Zubiaga** (2021). Towards generalisable hate speech detection: a review on obstacles and solutions. *PeerJ Computer Science*, **7**, e598.

423. **Yoder, M.**, **L. Ng**, **D. W. Brown**, and **K. Carley**, How hate speech varies by target identity: A computational analysis. *In Proceedings of the 26th Conference on Computational Natural Language Learning (CoNLL)*. Association for Computational Linguistics, Abu Dhabi, United Arab Emirates (Hybrid), 2022. URL https://aclanthology.org/2022.conll-1.3.

424. **Yu, S.**, **G. Da San Martino**, **M. Mohtarami**, **J. Glass**, and **P. Nakov**, Interpretable propaganda detection in news articles. *In Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2021)*. INCOMA Ltd., Held Online, 2021. URL https://aclanthology.org/2021.ranlp-1.179.

425. **Yu, W.**, **C. Zhu**, **Z. Li**, **Z. Hu**, **Q. Wang**, **H. Ji**, and **M. Jiang** (2022). A survey of knowledge-enhanced text generation. *ACM Comput. Surv.*, **54**(11s). ISSN 0360-0300. URL https://doi.org/10.1145/3512467.

426. **Zahrah, F.**, **J. R. C. Nurse**, and **M. Goldsmith**, A comparison of online hate on reddit and 4chan: A case study of the 2020 us election. *In ACM SIGAPP*. 2022.

427. **Zamfirescu-Pereira, J.**, **R. Y. Wong**, **B. Hartmann**, and **Q. Yang**, Why johnny can't prompt: How non-ai experts try (and fail) to design llm prompts. *In Proc. of the 2023 CHI Conference on Human Factors in Computing Systems*, CHI '23. Association for Computing Machinery, New York, NY, USA, 2023. ISBN 9781450394215. URL https://doi.org/10.1145/3544548.3581388.

428. **Zampieri, M.**, **S. Malmasi**, **P. Nakov**, **S. Rosenthal**, **N. Farra**, and **R. Kumar**, 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)*. Association for Computational Linguistics, Minneapolis, Minnesota, 2019*a*. URL https://aclanthology.org/N19-1144.

429. **Zampieri, M.**, **S. Malmasi**, **P. Nakov**, **S. Rosenthal**, **N. Farra**, and **R. Kumar**, SemEval-2019 task 6: Identifying and categorizing offensive language in social media (OffensEval). *In Proceedings of the 13th International Workshop on Semantic Evaluation*. Association for Computational Linguistics, Minneapolis, Minnesota, USA, 2019*b*. URL https://aclanthology.org/S19-2010.

430. **Zhang*, T.**, **V. Kishore***, **F. Wu***, **K. Q. Weinberger**, and **Y. Artzi**, Bertscore: Evaluating text generation with bert. *In International Conference on Learning Representations*. 2020. URL https://openreview.net/forum?id=SkeHuCVFDr.

431. **Zhang, Y.**, **V. Hangya**, and **A. Fraser**, LLM sensitivity challenges in abusive language detection: Instruction-tuned vs. human feedback. *In Proceedings of the 31st International Conference on Computational Linguistics*. Association for Computational Linguistics, Abu Dhabi, UAE, 2025. URL https://aclanthology.org/2025.coling-main.188/.

432. **Zhang, Y.**, **S. Nanduri**, **L. Jiang**, **T. Wu**, and **M. Sap**, BiasX: "thinking slow" in toxic content moderation with explanations of implied social biases. *In Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Singapore, 2023. URL https://aclanthology.org/2023.emnlp-main.300.

433. **Zhang, Y.**, **P. Tiňo**, **A. Leonardis**, and **K. Tang** (2021). A survey on neural network interpretability. *IEEE Transactions on Emerging Topics in Computational Intelligence*, **5**(5), 726–742.

434. **Zhao, J.**, **T. Wang**, **M. Yatskar**, **V. Ordonez**, and **K.-W. Chang**, Gender bias in coreference resolution: Evaluation and debiasing methods. *In Proceedings of the

2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers). Association for Computational Linguistics, New Orleans, Louisiana, 2018. URL https://www.aclweb.org/anthology/N18-2003.

435. **Zhao, Z.**, **E. Wallace**, **S. Feng**, **D. Klein**, and **S. Singh**, Calibrate before use: Improving few-shot performance of language models. *In Proc. of the 38th International Conference on Machine Learning*, volume 139 of *Proc. of Machine Learning Research*. PMLR, 2021. URL https://proceedings.mlr.press/v139/zhao21c.html.

436. **Zhou, F.**, **X. Xu**, **G. Trajcevski**, and **K. Zhang** (2020). A survey of information cascade analysis: Models, predictions and recent advances. *ArXiv*, **abs/2005.11041**.

437. **Zhou, K.**, **K. Ethayarajh**, and **D. Jurafsky**, Richer countries and richer representations. *In Findings of the ACL: ACL 2022*. ACL, Dublin, Ireland, 2022. URL https://aclanthology.org/2022.findings-acl.164.

438. **Zhou, X.**, **M. Sap**, **S. Swayamdipta**, **Y. Choi**, and **N. Smith**, Challenges in automated debiasing for toxic language detection. *In Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*. Association for Computational Linguistics, Online, 2021*a*. URL https://aclanthology.org/2021.eacl-main.274/.

439. **Zhou, X.**, **Y. Yong**, **X. Fan**, **G. Ren**, **Y. Song**, **Y. Diao**, **L. Yang**, and **H. Lin**, Hate speech detection based on sentiment knowledge sharing. *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)*. Association for Computational Linguistics, Online, 2021*b*. URL https://aclanthology.org/2021.acl-long.556.

440. **Zhou, X.**, **H. Zhu**, **A. Yerukola**, **T. Davidson**, **J. D. Hwang**, **S. Swayamdipta**, and **M. Sap**, COBRA frames: Contextual reasoning about effects and harms of offensive statements. *In Findings of the Association for Computational Linguistics: ACL 2023*. Association for Computational Linguistics, Toronto, Canada, 2023. URL https://aclanthology.org/2023.findings-acl.392.

441. **Zhu, W.**, **H. Liu**, **Q. Dong**, **J. Xu**, **S. Huang**, **L. Kong**, **J. Chen**, and **L. Li** (2023). Multilingual machine translation with large language models: Empirical results and analysis.

“If I must die

let it bring hope

let it be a tale.”

-Rafaat Alareer; If I must die