Prompt-GAN—Customisable Hate Speech and Extremist Datasets via Radicalised Neural Language Models

A thesis

submitted in partial fulfilment

of the requirements for the Degree

of

Master of Cyber Security (MCS)

at

The University of Waikato

by

Jarod Govers

THE UNIVERSITY OF

WAIKATO

Te Whare Wānanga o Waikato

2022

Abstract

Online hate speech and violent extremism knows no borders, no political boundaries, no remorse. Researchers face an uphill battle to collect hate speech data in volumes and topical diversity suitable for training state-of-the-art content-moderation systems. Neural language models ushered in a new era of synthetic data generation in use across various businesses, all despite calls for research to protect against unintended toxic output. We present a method for radicalising pre-trained neural language models to identify real online hate speech, as well as present the risks of rouge *radicalised* AI bots which could undermine our trust in social media. We present Prompt-GAN, a prompt-tuning adversarial approach with three achievements. We demonstrate prompt-tuning's ability to generate realistic types of hate *and* non-hate speech which mimics political extremist discourse. Prompt-GAN's architecture offers a twofold reduction in memory and runtime requirements compared to fine-tuning. Finally, Prompt-GAN improves hate speech classification F1-scores by up to 10.1% and sets a new record in neural language simulation compared to the current state-of-the-art across three benchmark datasets.

Acknowledgements

Thank you to my supervisors: Dr Panos Patros from the University of Waikato NZ; Dr Phil Feldman and Aaron Dant from ASRC Federal USA. Your collective help was instrumental in growing the research area of Extremism, Radicalisation, and Hate speech (ERH) context mining from a small project to a holistic research ecosystem. I am incredibly grateful for our discussions, advice and the ability push this project in my half-research/half-course Master of Cyber Security, to the level of outcomes expected of a research-only Master's. Thank you for growing and fostering my passion in pushing the state-of-the-art, as well as addressing the societal challenges shared between our countries in these unprecedented times.

Contents

Chapter 1

Introduction

While the use of online social media to radicalise individuals is a core focus for the War on Terror, the 2021 Capitol hill riot/insurrection and the 2019 Christchurch mosque shootings acted as a wake-up call for the bias towards Islamic extremism by social media content-moderation and intelligence communities [92, 39]. Our prior systematic literature review identified a disproportionate bias towards counter-extremism research on Islamic extremism [39]. Whereby, 24% of United States originating Extremism, Radicalisation, and Hate speech (ERH) studies targeted Islamic extremism compared to only 10% for far-right extremism—despite 90% of terrorist attacks in the US being from far-right extremists in 2020 [47]. Compounding these factors, international developments such as the COVID-19 pandemic, and the Russo-Ukrainian war offer avenues for state and non-state actors to seed disinformation and polarising rhetoric to undermine democratic institutions through deliberate polarisation.

All forms of social media will have some form of content-moderation policy—ranging from normative consensus for issues such as countering child exploitation or violent threats, to more disputed realms of non-violent but discriminatory hate speech [31, 97, 15]. Platforms such as Facebook and Twitter continue to counter hate speech and the use of their platforms by terrorist organisations [31, 97]. However, mainstream platforms such as Twitter and YouTube have algorithmic biases towards recommending polarising and extremist content due to the notoriety and click-worthy nature [45, 56].

Surrounding social media are the challenges to online privacy and researcher safety for collecting real-world hate speech datasets, as well as human biases in labelling *Extremist* affiliation, *Radicalisation* processes, or *Hate speech* (ERH). In essence, these issues transcend disciplines. Hence, this study acts as a praxis between sociolinguistics, artificial intelligence, and cyber-security.

For sociolinguistics, human annotation requires various interpretations of content-moderation policies. Commonly, researchers utilise inter-rater agreement between a panel of 3-5 participants to vote on whether a flagged post violates community guidelines [39]. Understanding how hate perpetuates is as much a social task as it is one of language and systematic processes.

Artificial intelligence can automate the content-moderation process to identify hate speech and extremism among large volumes of online data. Thus, Prompt-GAN relies on the computer science field of Natural Language Processing (NLP) via text generation and classification with deep learning models.

Cyber-security research often requires manipulating existing systems in search for threats and vulnerabilities. Ethical 'white hat' hacking is not antithetical to online safety, as researchers create deliberate attacks to identify weaknesses and vulnerabilities within a controlled environment to create preemptive solutions. If we understand how a malicious 'hate or non-hate bot' operates and its similarities to human speech, then ethical researchers can also derive mitigations against malicious attackers who intend to cause harm through *automating* hate through online bots.

Simulating online hate invokes evocative fears of rogue AI from the fictitious Skynet to real-world fault lines—as seen by a mental health AI which told a simulated patient to commit suicide [22]. There is a lot to learn from both how AI 'think' and the patterns of human discourse that AI can mimic. What models like the Generative Pre-trained Transformer (GPT) output can also offer a reflection of the global biases, viewpoints, and memetic culture [1, 14].

Likewise, customised bots can intentionally destabilise online culture as a form of modern information warfare. The United States Intelligence Community reported that the Russian private company, the Internet Research Agency, acted as an organised group of "professional trolls located in Saint Petersburg... [and acted as] a close ally of Putin with ties to Russian intelligence" [101][p. 4]. While including paid employers to act as online trolls to perpetuate hate, Ukrainian intelligence outlined the increased use of Russian black-box bots to provoke violence against the democratic Ukrainian government to destabilise and instil a civil war [87]. The architecture for such state-actor bots is shrouded in classification and secrecy, limiting our understanding to prevent such information warfare. Thus, open cyber-security research into the use of AI for malicious purposes is essential to identify patterns and construct benchmark datasets to classify and identify *synthetic* radicalisation. In essence, synthetic online text generating bots enables researchers to improve content-moderating *and* anti-bot systems to counter the new era of online psychological operations by state actors and identify radicalising linguistic patterns.

1.1 Motivation

Collecting hate speech datasets invokes three core challenges:

1. The data must be up to date, to reflect new events and online culture.
2. The process of human annotation of online data is both time-consuming and costly—with researchers outsourcing annotation to paid services such as Amazon Mechanical Turk or CrowdFlower [39].
3. Collecting online data for persistent datasets undermines online privacy rights—particularly a user's right to be forgotten. Even in publicly-accessible posts, a user would likely object to having their data on record indefinitely and be labelled as an 'extremist'.
4. Data collection raises risks to a researcher's safety like their mental health when annotating real data, and retribution from collected users [19, 66].

Hence, the motivation for our Prompt-GAN model is to reduce the time, privacy risks, and enhance research safety through synthesising online discourse using deep neural language models. We also seek to conduct an exemplar of AI stress testing and *social* vulnerability discovery of public-facing text-generation systems such as GPT-2/Neo—to analyse how an attacker could exploit a pre-trained model without manipulating the underlying model weights or code. With GPT-3 in commercial use [75], it is essential to understand whether such models are actually safe to use for customers—particularly given the risks of a malicious cyber-attacker using radicalising prompts to incite hatred to an unsuspecting customer base.

Furthermore, we seek to expand on the field of prompt-tuning of large (over a billion parameter) language models to reduce the memory and runtime requirements to generate synthetic text. As neural language model parameter counts scale exponentially, the financial and environmental cost of training may not justify plateauing realism improvements—with GPT-3 costing ~$12 million to train with ~85,000 kg of CO2-equivalent emissions [79]. While our task focuses on hate speech generation and classification, Prompt-GAN's architecture is compatible with any open-ended real-vs-fake textual task, such as question-answering, chatbots, journalism, fact-checking, among other open-ended tasks. Prior prompt-tuning work explore similar tasks like question-answering [55, 62, 112], but is not the focus of this hate speech study.

1.2 Contributions

Thus, our core contributions of our novel Prompt-GAN model are:

1. An effective string-builder approach for prompt-tuning through a Generative Adversarial Network (GAN) approach to text generation—decreasing memory and runtime requirements for large text language models.

2. A new approach to dataset generation to reduce the cost and time to collect and annotate social media data.

3. An investigative framework for analysing textual realism through a *digital Turing test*—which demonstrates Prompt-GAN's effectiveness in creating realistic text in topics both covered in the dataset, and out-of-corpus topics through our *domain expansion* approach.

1.3 Thesis Structure

To contextualise Prompt-GAN, we present a summarised *'algorithm handbook'* to machine learning and social media definitions in our *Background* Chapter 2. Both the background and literature review chapters include summaries and excerpts from our full under-review 35-page Systematic Literature Review on online *Extremism*, *Radicalisation*, and politicised *Hate speech* (ERH) [39]. We also include a literature review on text generation techniques in Section 3.2.

Thereafter, our *Research Design* Chapter 4 presents the three components of Prompt-GAN, consisting of the automated prompt-builder model, the GPT-2/Neo local text generator component, and text discriminator component.

We continue with the evaluation of Prompt-GAN's data in Chapter 5. Our results include metrics from our baseline hate classification models', including the use of our synthetic posts to supplement or replace the real training data.

Chapter 6 offers a discussion of the *digital Turing tests* on our synthetic data, including its utility to create out-of-corpus topics and future hate speech datasets via our *domain expansion* approach.

We conclude with our recommendations for future work in industry, government, and academia in our *Conclusions and Future Work* Chapter 7.

For a quick synopsis of this study, refer to our visualisation of Prompt-GAN's architecture in Figure 4.1 in our *Architectural Design* Section 4.2; alongside the *Results* Section 5.2, which outlines Prompt-GAN's improvements to the classification performance on the real datasets. Thereafter, continue to the summary of our three research questions and its implications and recommendations for future work in the *Conclusions and Future Work* Chapter 7.

Chapter 2

Background

Analysing social media requires the socio-technical considerations of what constitutes *Hate speech*, *Extremism*, and *Radicalisation* (ERH). Hence, this chapter decouples and analyses ERH's social background and definitions, which are essential concepts to define for criteria-building for our synthetic datasets. Section 2.2 outlines the technical definitions and algorithmic approaches for online ERH detection, as well as the methods for generating synthetic text to mimic human syntax, structure, topics, and speech. We conclude with a framework for analysing online hate speech through our evidence-driven proposed research area of ERH Context Mining—first outlined in our separate interdisciplinary systematic literature review on ERH concepts [39].

2.1 Context to Social Network Analysis

To identify online hate speech and extremist organisation affiliation, computational models can investigate text, audio-visual and network sources—including textual meaning and intent through the field of *Natural Language Processing (NLP)*, *computer vision* for visual sources, and evaluating relationships between users, themes, or beliefs through *community detection*. The amalgamation of textual *NLP*, *community detection*, and content recognition via *multimedia computer vision* thereby constitutes the three pillars to our proposed research area of *ERH Context Mining*, which builds on the algorithmic analysis of language and society enshrined within the *computational sociolinguistics* field. Hence, this section relies on the international academic consensus to decouple the baggage of what constitutes hate speech, extremist affiliation, and radicalisation towards extremist ideologies.

2.1.1 Extremism and Radicalisation Decoupled

Extremism's definition appears in two main flavours: politically fringe belief systems outside of the social contract, such as those represented by small political parties within the political system; or violence supporting organisational affiliation which seeks to violently overthrow or reform government(s) to implement policies through terror.

The Anti-Defamation League (ADL) frames extremism as a concept "used to describe religious, social or political belief systems that exist substantially outside of belief systems more broadly accepted in society" [4]. Under the ADL's definition, extremism can be a peaceful *positive* force for mainstreaming subjugated beliefs, such as for civil rights movements. This construct of a socially mainstream belief constitutes the *Overton window* [64]—the range of socially acceptable political views, which varies by country, culture, and time. Content-moderation is not a tool for targeting political affiliation, but to target violent and/or hateful speech and actions—regardless of its position on the Overton window and wider political spectrum.

Conceptually, extremism typically involves hostility towards an apparent 'foreign' group based on an opposing characteristic, circumstance or ideology. Extremism often relies on defending and congregating people(s) around a central ideology, whose followers and devotees are considered *in-group* [13, 10]. Extremists unify through hostility and a perceived injustice from an *out-group* of people(s) that do not conform to the extremist narrative—typically in a "us vs. them" manner. Hence, extremism detection algorithms can use non-textual relationships as an identifying factor via clustering users into communities (i.e., *community detection*). Thus, extremism can simply reduce to *any* form of a fringe group whose identity represents the vocal antithesis of another group. Thereby, extremism can indicate new groups and movements, and does not have to be a focus for content-moderation.

Facebook, Twitter, YouTube, and the European Union frame extremism as a form of indirect or direct support for civilian-oriented and politically motivated *violence* for coercive change [29]. Facebook expands this industry-wide consensus to include *Militarised Social Movements* and "violence-inducing conspiracy networks such as QAnon" [31].

Radicalisation focuses on the process of ideological movement towards a different belief space, which the EU frames as a "*phased and complex process* in which an individual or a group embraces a radical ideology or belief that accepts, uses or condones violence" [30]. An instrument of violence-supporting political action used by violent extremists are acts of terrorism. Whereby, terrorism consists of politically motivated violence towards the civilian population to coerce, intimidate, or force specific political objectives, which is an end-point for violent *radicalisation* to project *extremist* ideology.

Nonetheless, ERH detection does not offer a panacea to combating global terrorism, nor does surveillance offer a 'catch-all' solution. In the case of the livestreamed Christchurch shooter (an extremist via his desire to use violence to push political change), the New Zealand Security Intelligence Service concluded that "the most likely (if not only) way NZSIS could have discovered [the individual]'s plans to conduct what is now known of [the individual]'s plans to conduct his terrorist attacks would have been via his manifesto." [92, p. 105]. However, the individual did not disseminate this manifesto online until immediately before the attack, and his 8Chan posts did not pass the criteria to garner a search warrant [92, p. 105]. Hence, extremism detection is an evolutionary arms race between effective and ethical defences vs. new tactics to evade detection.

2.1.2 Hate Speech Decoupled

The European Union defines hate speech as all conduct “publicly inciting to violence or hatred directed against a group of persons or a member of such a group defined by reference to race, colour, religion, descent or national or ethnic origin.” [29, p. 1]

The U.S. Department of Justice has its own definition of a hate crime as “a traditional offence like murder, arson, or vandalism with an added element of bias... [consisting of a] criminal offence against a person or property motivated in whole or in part by an offender’s bias against a race, religion, disability, sexual orientation, ethnicity, gender, or gender identity.” [32]

The New Zealand Government frames hate speech in the context of persons “who, with intent to excite hostility or ill-will against, or bring into contempt or ridicule, any group of persons in New Zealand on the ground of the colour, race, or ethnic or national origins of that group of persons” under Section 131 of the Human Rights Act 1993 [44]. Notably, governmental laws may differ from industry content-moderation policies via the omission of sexual, gender, religious or disability protections, and may protect non-violent but insulting speech.

Online US-based platforms such as Facebook and Twitter are free to determine the rules for their content-moderation policies on their platform—including restricting access for discourse. The First Amendment enshrines these platforms’ ability to determine what is and isn’t allowed on their platform, as ruled in cases such as Zhang v. Baidu [100] and via law with Section 230 of Title 47 of the U.S. Code [99, 27].

The United Nations Office on Genocide Prevention and the Responsibility to Protect outlines the international consensus on hate speech as “any kind of 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 factor.” [98, p. 2]

What these definitions have in common is that they all involve speech directed at a portion of the population based on a protected class. A protected class may include, but is not always limited to, race, sex, gender or sexual orientation, origin, or disability status [4, 98, 29, 32, 44]. For example, "All black people are criminals" would be hate speech as it attacks an entire group of people based on their skin colour—with a negative intention due to a demeaning or dehumanising intent. Most U.S.-based hate speech laws involve attacking individuals based on their membership in a protected class.

For the purposes of Prompt-GAN, we focus our experimental design on generating types of hate speech due to its focus on textual natural language processing. Furthermore, related text generation approaches of fine-tuning and Long Short-Term Memory (LSTM) models rely on binary hate speech datasets—which we identified as common "benchmark datasets" in Section 3.1.

We define hate speech for the purposes of Prompt-GAN as "Targeted, harassing, or violence-inducing speech towards other members or groups based on protected characteristics."

2.2 Technical Definitions for Text Generation and Classification via Deep Learning

Deep learning represents a family of machine learning algorithms based on neural networks with typically three or more nested layers. Neural networks rely on training a network of interconnected nodes which receive signals from other nodes in the network. Values across the network, known as weights, change via non-linear functions to process and alter the value to send to surrounding neurons for further processing. Hence, these nodes 'mimic' the brain's biological neurons in this neural network. For instance, the first layer of a neural network utilises a numeric representation of arbitrary data, such as numeric values to represent parts of, or entire words. These numeric 'tokens' are adjusted throughout the hidden lower layers towards a final output layer for classification or generation tasks. Adjusting the weights such that the input values traverse down towards an output decision layer to a desired output (e.g., *hateful* text generation) constitutes *training* the neural network.

Furthermore, each downwards training step results in readjusting the weights of the upper layers of neurons—through a process known as *backpropagation* [105]. Figure 2.1 displays this architecture for neural networks per our example of hate speech detection via Natural Language Processing (NLP). The benefit of deep learning in ERH detection is the preservation of word order and meaning (e.g., "I ran a marathon" vs "I ran for president"), thus displaying context-*sensitivity.* Given dual-use words such as "queer", or racially motivated slurs, understanding the surrounding contextual words is essential to reduce classification bias [71].

2.2.1 Transformer Language Models

The state-of-the-art transformer architecture relies on *self-attention*—the memory retention of neural networks where each token in a sequence is differentially weighted [25, 14]. Unlike Recurrent Neural Networks, where the neural network's nodes follow a temporal sequence, a transformer's attention mechanism utilises context for any position in the token sequence. Hence, transformers can handle words out of order to increase textual performance in text generation or classification tasks. Transformers offer greater classification performance compared to non-neural network machine learning approaches, as identified in our literature review chapter through Table 3.3. However, while non-deep learning algorithms such as Naïve Bayes can require as little as one equation to calculate a classification probability, neural networks can involve millions to billions of operations per the number of neurons in the network, increasing classification/model performance at the expense of memory and computational overhead. A considerable ethical threat of transformer models is their ability to predict future tokens (i.e., text generation). For instance, a malicious actor could create realistic automated trolls or radicalising synthetic agents as bots. Language models also risk data leakage of their trained data through predicting tokens found in the original trained dataset, such as names or addresses [14]. Hence, while our model relies on public Open-Source Intelligence datasets pulled from publicly accessible tweets and forum posts, we do not open-source our hate speech generating code to prevent malicious use.

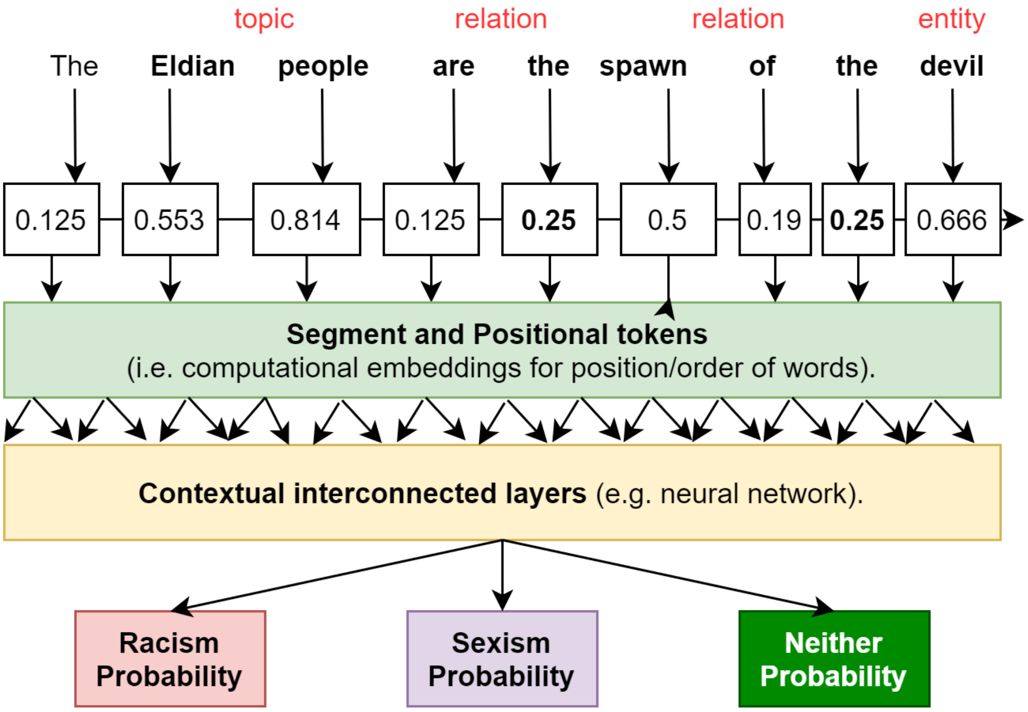


Figure 2.1: An abstracted example of a neural network for text classification. The top text represents its raw syntactic form with its converted numeric embedding representation. These embeddings are responsible for altering the weights to increase token prediction or generation (for transformers) via back-propagation. The final output layer for this example would offer the probability that the given text is racist, sexist, or benign. N.B. Eldians are a fictitious persecuted race as depicted in the fantasy series, *Attack on Titan*.

2.2.2 Learning Language Models—Defining Training, Fine-tuning, and Prompt-tuning

There are three key strategies to generate a desired output from language models. Firstly, learning patterns to generate a desired output from all zero or randomly initialised weights constitutes full training. Training from scratch requires a significant corpus of data to ensure that the neural network understands complex human topics, entity-event relationships, themes and facts; as well as linguistic structure and syntax for the training dataset(s) language. For the three language models we consider, the pre-trained corpus includes:

1. The training data used to train the Robustly Optimized Bidirectional Encoder Representations from Transformers (RoBERTa) model—pre-trained on entries from English Wikipedia (2.5 million words), the English BookCorpus (800 million words), 63 million news articles, forums, and stories across the 125 million parameter roberta-base model [61].

2. GPT-2's training data—pre-trained on 40GB of internet text varying from Wikipedia to online forums and websites scrapped using a custom in-house web scrapper [80].

3. GPT-Neo training data—pre-trained on 886.03GB of online text ranging from one-third academic sources (PubMed Central, ArXiv, FreeLaw among others), web text (CommonCrawl, OpenWebText2 on Reddit and Twitter, Wikipedia, StackExchange), book corpora, code, subtitles and comments from YouTube, and open-source public forums [38].

Secondly, we can reduce the amount of data to process by instead modifying the existing weights to bias a pre-trained model towards generating a desired output for classification or text generation. Whereby, *fine-tuning* is a form of transfer learning—utilising the original source model's weight and freezing a subset of layers. Typically, freezing the pre-trained weights are done across whole layers in between the input and output layers. As the pre-trained model already understands general linguistic and epistemological patterns, fine-tuning a BERT-based model can require as little as adding a single output layer for classification tasks [25]. Fine-tuning more layers will result in higher classification performance than fine-tuning with more frozen/fixed layers. For instance, BERT's question-answering F1-score performance (i.e., a harmonised value balancing accuracy and precision) remains stable when freezing the first eight layers but deteriorates when freezing any of the last four layers [34]. Unknowns in working with BERT includes determining the ideal performance-runtime trade-off between the number of layers to freeze for fine-tuning and how language models respond to new patterns. Collectively, understanding the decisions made by BERT-based models underpin a 'unknown unknown' in machine learning in a field framed by Rogers et al. as BERTology [85].

Another issue with fine-tuning is that it requires altering the weights within the network to instead bias the input dataset. For instance, fine-tuning a GPT-2 model on a hate speech dataset can generate hate speech which can supplement a real-world dataset, as demonstrated by the MegaSpeech corpus consisting of one million synthetic sentences from a fine-tuned GPT-2 model on a corpus of hate speech data [107]. A pre-trained GPT-2/3 model will have a generalised understanding of geopolitical events and entities due to its pre-trained corpora consisting of Wikipedia, BookCorpus, and academic sources [14, 12, 80]. However, the fine-tuned model sacrifices its knowledge of its pre-trained data to instead mimic the fine-tuned dataset—causing the new model to forget its previously learned latent information. Hence, a fine-tuned model on a white supremacy US-centric hate speech corpus is unlikely to generate reliable hate speech for anti-White racism or non-hate speech instances due to the *catastrophic forgetting* of its pre-trained knowledge base.

In the example below, we demonstrate the issue of catastrophic forgetting of useful hate-specific information by comparing GPT-2 and a fine-tuned (on the de Gibert et al. [23] hate speech dataset) GPT-2 model:

The corresponding outputs in Table 2.1 reflect this bias towards replicating the fine-tuned hate corpus, in turn losing its prior knowledge of events—typically resorting to blaming any X event as being a good or bad thing for the targeted racial class. This catastrophic forgetting of its original social knowledge is detrimental if we want to generate diverse multi-topic and multi-ethnic datasets. Specifically, querying entities of any form tend to generalise anti-

Table 2.1: Output from prompt: *“Question: What is the September 11 attacks? Answer: ”*

GPT-2 (1.5B) Baseline	GPT-2 Fine-tuned on de Gibert et al. hate corpus
“The September 11 attacks were attacks on the United States by al-Qaida and Osama bin Laden.”	*“The attacks are an example of Africans.”*

Jewish or anti-black hate speech, often claiming that any event is to target the perceived targeted “white race”. Thus, we cannot create opposing speech, or hate speech from other ethnic, political, or social perspectives (e.g., Islamic extremist speech which is more likely to target the white demographic of the de Gibert et al. Stormfront data [23]). Hence, we postulate that language models’ diverse social and political knowledge, paired with its online posts within its pre-trained corpus, will all suffice for hate speech generation without the need for fine-tuning.

2.2.2.1 Prompt-tuning

Thus, a third way to create synthetic data without further training is through prompt modification. Given the input layer for language models such as GPT, we can modify the input prompt to provide an instruction with the intention to only modify the input ‘prompt’ layer, and not manipulating any of the model’s inner layers/weights (unlike fine-tuning or full training, which relies on modifying the model). Prompt-tuning can rely on manual trial-and-error to identify prompts that create *consistent*, *repeatable*, and *realistic* outputs [55, 112, 62]. As an alternate to manipulating textual prompts through manual design, Lester et al. [55] considered perturbing numeric embeddings for the T5 model using a pre-existing crafted prompt to enhance model performance on a multitude of text generation tasks. However, both approaches require manual trial-and-error to create a starting prompt. The key benefit to prompt-tuning is that it retains the model’s original knowledge from its pre-trained corpus, as the weights are not modified to generate a biased output (e.g., fine-tuning to create *hateful* output posts)—which is the cause for catastrophic forgetting.

However, modifying prompt *embeddings* presents two challenges:

1. Modifying the input-layer embeddings requires modifying up to 1024 dimensions—leading to a feature explosion due to the embedding space's high dimensionality.

2. Given the highly specific task for prompt-tuning, most of the embedding space will not be useful for the task or even linguistically relevant.

Embeddings which reflect seemingly random prompt strings will create random output. Specifically, generating a prompt input must follow a linguistic structure and understand GPT's embedding space, as well as understand the niche task to solve.

Thus, prompt design and soft prompt manipulation currently require a static starting input prompt or embedding vector to generate text which guides the model towards the topic/task required. Google's trainable soft-prompts are also limited to only 20 tokens, due to the high dimensionality exponentially increasing performance (as each token is contextual, thus changing one token's embeddings would impact all) [55]. To our knowledge, no studies consider automated means of reducing the prompt token selection space (i.e., avoiding embeddings which are not relevant to the task or topic) or instruction-tuning. Likewise, Lester et al. designed their 20-token soft prompt vector model around the T5 language model's encoder-decoder architecture, which is incompatible with GPT's faster decoder-only architecture [55, 67]).

Prompt-tuning can reduce memory requirements by at least 50%, as fine-tuning requires a copy of the optimiser weights and activations for further processing via backpropagation. As prompt-tuning does not alter any weights, runtime for text generation is conceptually faster as only forward operations (i.e., traversing the neural network, known as *inference*) are required to generate the desired output, with no need to store or process any intermediate weights. In our example of catastrophic forgetting in Table 2.1, inference on the 345 million parameter GPT-2-medium model required 3.56GB of Video Random Access Memory (VRAM), while fine-tuning the model required 8.38GB for the GPT-2-medium. Thus, prompt-tuning further enables us to test larger language models with more realistic text generation capabilities before running out of available memory. With our Prompt-GAN architecture, we can reduce the training time to be ~3.5x faster than fine-tuning—26 minutes to train via inference with Prompt-GAN, compared to 92 minutes to fine-tune the same size GPT-2-XL model on the same Stormfront dataset.

2.2.2.2 Cross-encoders (e.g., BERT)

Bidirectional Encoder Representations from Transformers (BERT) is the most common cross-encoder observed for ERH detection [25]. BERT upholds the highest performance of all NLP models observed in our literature review as they outperform non-deep approaches by 10% by F1-score [39]. Cross-encoders offer higher performance for classification tasks, through retaining information over a given sequence with a label (i.e., self-attention). BERT's strength is its memory retention of all tokens in a sentence, thus upholding full context-sensitivity of every word in the input text. However, cross-encoders are computationally expensive due to their high parameter counts (110 million parameters for BERT-base, 365 million for BERT-large). Hence, an area of ongoing research includes model distillation, which consists of optimising and reducing a model's parameter count by training a lower parameter count model to predict the patterns of a larger model's weights to reduce memory requirements and training time [42]. Other methods for optimising BERT classification performance include increasing dataset size, adding multi-class labels to provide BERT more diversity for training, and alternate layer architectures [61, 84, 109].

2.2.2.3 Generative Pre-trained Transformer (GPT)

Similarly, the state-of-the-art GPT transformer architecture expands on the encoder blocks (shared with BERT) to include decoder blocks [14]. Hence, GPT works on a token-by-token basis by estimating a sequence's next token—ideal for tasks such as text generation, summarising, question answering, and information retrieval.

GPT models differ from BERT-based models via *masked self-attention*—an alternate form of context-sensitivity where the model only knows the context of the prior words in the sentence. GPT-2/3 [14], GPT-Neo [12], and Jurassic-1 [57], are notable 2019-2022 era multi-billion parameter models—where their larger pre-trained corpus and parameter count result in generating better performance in information retrieval and text generation tasks [14].

2.2.2.4 Siamese and Triplet BERT (Sentence-level BERT)

Sentence-level BERT extends the BERT model through generating two parallel contextualised word embeddings—and pooling the output for the overall sentence [84]. The core rationale for SBERT over BERT is its utility for finding the most similar pair of sentences, known as Semantic Textual Similarity (STS). Calculating the most similar sentence among a collection of 10,000 sentences using the word-by-word BERT embeddings can require up to 50 million inference operations. Hence, Reimers and Gurevych [84] proposed pooling the embeddings from two sentences and utilising cosine similarity across the pooled sentence embeddings to reduce the computational complexity of STS tasks. We consider SBERT in our experimental design to identify if a pre-trained SBERT model can detect the semantic differences between hate and non-hate data—which we investigate and visualise in our Evaluation Chapter 5.

As the SBERT architecture is a wrapper consisting of a triplet network of BERT models, we consider and substitute multiple types of BERT models when identifying the semantic similarity between hate and non-hate data. Table 2.2 presents the four models we select for the SBERT network as based on their state-of-the-art sentence embedding and semantic search performance [83]. To balance runtime, we also consider a distilled version of the SBERT model for computational, memory and energy efficiency.

We hypothesise that contextual sentence-level embeddings could help generate new sentences with high semantic similarity to hate speech data. If this hypothesis holds, then we could use the semantically similar hate sentences as a starting prompt for the GPT-2/Neo synthetic post generator.

Model	Performance Sentence Embeddings	Performance Semantic Search	Average Performance	Speed
All-mpnet-base-v2	69.57	57.02	63.3	2800
All-roberta-large-v1	70.23	53.05	61.64	800
All-distilroberta-v1	68.73	50.94	59.84	4000
Average word embeddings glove.6B.300d	49.79	22.71	36.25	34000

Table 2.2: Comparison of SBERT variants [83]. Performance for sentence embeddings average across 14 STS datasets, while semantic search performance is average across 6 datasets. Average performance aggregates all 20 datasets.

2.2.3 Generative Adversarial Networks (GAN)

Generative Adversarial Networks (GANs) reflect a machine learning framework consisting of two key components: a generator, and a discriminator neural network. The principle of a GAN is to generate synthetic data and discriminate the synthetic data from real data. The relationship between the synthetic data generator (typically a separate neural network) and the discriminator network reflects a zero-sum game, whereby the generator's training objective is to create synthetic data that maximises the loss function of the discriminator—in effect, aiming to "trick" the discriminator network into classifying the synthetic data as real, and real data as fake. After the generator network's training run (*epoch*), the discriminator network retrains using the new synthetic data, along with the original data—aiming to learn the new patterns between real and fake. The initial state of the generator is typically a randomly generated latent space which is then trained towards the embedding space of the desired output. To ensure consistency in training, the GAN relies on training the generator and discriminator networks separately—either testing the new weights/setup for the generator or freezing the generator's weights and training the discriminator. Figure 2.2 displays the full architectural process of training the adversarial generator to training the discriminator and offering feedback to the generator via backpropagation or a policy update.

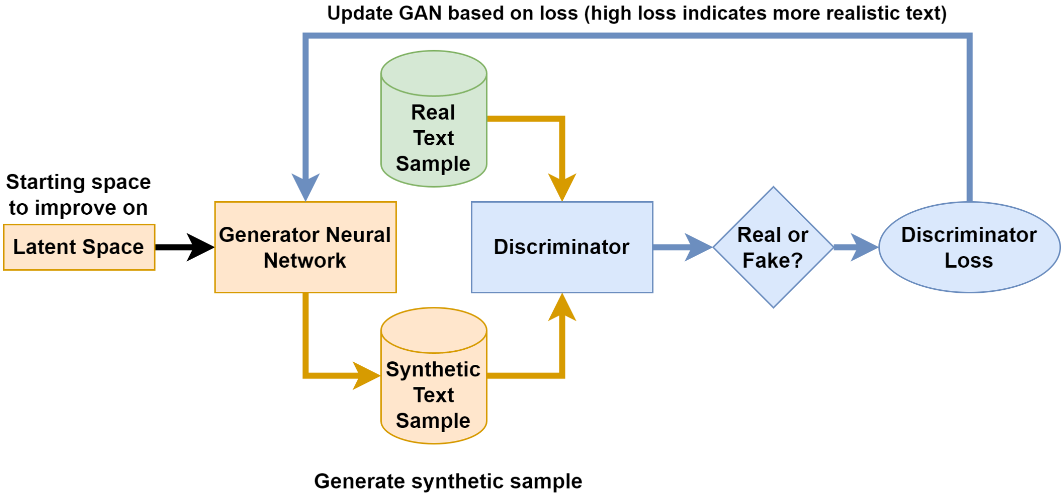

Update GAN based on loss (high loss indicates more realistic text)
Real Text Sample
Starting space to improve on
Latent Space
Generator Neural Network
Discriminator
Real or Fake?
Discriminator Loss
Synthetic Text Sample
Generate synthetic sample

Figure 2.2: Architecture for a generic Generative Adversarial Network (GAN).

For instance, the below results from the image generating StyleGAN2 face generator would initially create static white noise images which would be easily distinguishable from the real-life faces in the training dataset [50]. However, the perturbed generator's neural network weights will learn the training data's shapes, colours and patterns. Initially, this may be as simplistic as creating an image with a circle, given that human heads are typically circular in shape. If the generator tests a change in weights that results in less human like qualities (e.g., testing a triangle shaped head), this will result in a lower discriminator loss, and thus the generator would reject the policy changes. Training ceases after a predetermined amount of generator and discriminator epochs, or when the discriminator's loss stabilises. The final result demonstrates compelling human-like synthetic faces, as displayed with three examples from StyleGAN2 in Figure 2.3.

Figure 2.3: Three synthetic faces generated via Nvidia's StyleGAN2 model [50].

In the context of Prompt-GAN, our aim is to optimise the arbitrary textual input to GPT-2/Neo to create synthetic text to trick a BERT-based discriminator. Our intention is that such arbitrary text reflects different forms of hate and non-speech as stylised towards the target platform—specifically tweets of up to 280 characters, and longer arbitrary-length posts stylised to mimic the extremist white supremacist forum Stormfront.

2.2.4 Definitions for Prompt-GAN's Feature Extraction Techniques

This subsection outlines the four feature extraction techniques used for textual ERH detection, consisting of numerical representations for word or entities via word-vector embeddings via Word2Vec [70] and Wikipedia2Vec [108], topic extraction and related topic search via BERTopic [40], and text frequency analysis techniques via Term Frequency-Inverse Document Frequency.

2.2.4.1 Embedding Representations and Word2Vec

To represent textual words in manner conducive to search and identify 'nearest/most similar' words, *Word2Vec*-based approaches represent words as an n-dimensional array. In a 2D array, one could imagine words with similar embeddings acting as coordinates on a map, whereby words like *chocolate* and *vanilla* sharing similar 'coordinate' embedding values and thus being closer together than non-food words like *house* or *apartment.* Calculating the spatial similarity of words represented as multidimensional vectors (i.e., embeddings) typically utilises cosine similarity, with the equation of:

$$cos(\theta) = \frac{x \cdot y}{\|x\|\|y\|} \tag{2.1}$$

Cosine similarity equation

Cosine similarity measures the cosine of the angle between the two multidimensional vectors. In equation 2.1, x and y reflect the numeric embedding arrays for two words. A cosine similarity value of 0 reflects that the multidimensional arrays share no similarity, as the output angle value are at an orthogonal 90-degree direction. Conversely, a cosine similarity of 1 demonstrates a matching angle between the vectors—reflecting that the two embeddings are identical. Where the overall embedding space reflects semantic similarity, a cosine similarity value closer to 1 reflects words of similar meaning.

Word2Vec is a model to convert words into vector embeddings, which compares synonymous words (e.g., *hate* and *disgust*) via numerical vectors [70]. On a word-level basis, the vector value for *king* minus the value for *man* and adding the vector value for *woman* would equal a vector similar to *queen* [70]. In our case, the concept of an *Islamist extremist* and *ISIS* are semantically similar akin to *White supremacy* and *Nazism.*

2.2.4.2 Wikipedia2Vec

Wikipedia2Vec extends the word-vector relationship of Word2Vec with a graph-based entity-relationship model [108]. In addition to linking words based on its use and context within the training corpus, Wikipedia2Vec's graph-based entity-relationship model works by linking the embeddings between the *terms* and the *articles* within Wikipedia—to link *term use* and *entities* (i.e., events, individuals, groups, and any Wikipedia page's title). The learned embeddings of similar words and entities cluster together within a 100-dimension embedding space and rely on three sub-models:

1. *Wikipedia link graph model*—an undirected graph where the nodes represent Wikipedia article titles (i.e., entities) and the edges represent links between entities throughout Wikipedia.

2. *Word-based skip-gram model*—similar to Word2Vec, whereby neighbouring words provide context to a target word's vector in the sparse embedding space.

3. *Anchor context model*—grouping words and entities together, where the model learns these embeddings by predicting the neighbouring words/terms for each entity.

The training process for these three sub-models rely on skip-gram training—which uses the nearest embeddings to predict contextual words and entities given a target term. Skip-gram training could include searching for related words and terms needed to correctly predict the target word *rugby*, which may include finding relevant rugby players and the overall topic entity of *sports*.

No studies consider Wikipedia2Vec for hate speech detection or generation. Our Prompt-GAN architecture explores Wikipedia2Vec to identify related concepts, entities, and terms to seed/prepend to our input prompt to generate diverse hate and non-hate data. Our approach includes automatically searching historical events and entities to create internationalised multi-political datasets—in a method we hybridise with multiple feature extraction techniques and henceforth frame as *'domain expansion'*.

To ensure the topical relevance of our synthetic data given the dynamic and rapidly changing nature of online discourse, we do not use the original April 2018 Wikipedia model. Instead, we retrain a 100-dimension model on the April 2022 Wikipedia corpus—and integrate both the link-graph, mention database of entities-referring-other-entities and textual embeddings to create a

3.5GB output model. Thus, the updated model will include topics, events, and notable individuals post-2018—enabling our Prompt-GAN model to explore topical radicalising events such as the ongoing COVID-19 pandemic. We port the updated model to PyTorch 1.11 and Python 3.10 for compatibility with our PyTorch and Huggingface compatible Prompt-GAN architecture.

2.2.4.3 Topic extraction via BERTopic

BERTopic is an embedding-based topic clustering approach which relies on encoding the datasets into a pre-trained BERT model——which converts the tokenised dataset into a 768-dimension topic-based embedding array [40]. As finding the nearest topics from 768-dimension is computationally expensive due to the embedding space's high dimensionality, BERTopic's architecture transforms the high-dimensional sparse embeddings to a lower-dimension manifold via Uniform Manifold Approximation and Projection (UMAP) for Dimensionality Reduction. Thereafter, BERTopic reduces the computational complexity of calculating the proximity of related word embeddings via clustering the lower-dimension approximations for the topic embeddings. Whereby, the BERTopic architecture employs the Density-Based Spatial Clustering of Applications with Noise (DBSCAN) dimensionality reduction technique to identify areas of high-density embedding clusters and related clusters to reduce an otherwise sparse dimension space. From these clusters, we can identify class-specific words by ranking the frequent and related topics via Maximal Marginal Relevance (MMR) to rank relevant keywords by its topical similarity.

2.2.4.4 Term Frequency-Inverse Document Frequency (TF-IDF)

TF-IDF determines the relevance of a word in a document by comparing its frequency *in the document* to its inverse number for the frequency of that word *across all documents* [105]. Thereby, assigning each word a weight to signify its semantic importance compared to the wider corpus. For instance, radical Islamist *dog-whistle terms* (i.e., coded or suggestive political messages intended to support a group) appeared disproportionately in extremist text compared to a neutral religious corpus [82].

The base equation for TF-IDF consists of:

$$\mathrm{tfidf}(t, d, D) = tf(t, d) \times idf(t, D)$$

Where in our case, we use a variant of TF-IDF with smooth normalisation/regularisation where the IDF component is:

$$\mathrm{idf}(t) = \log \frac{n}{\mathrm{df}(t)} + 1$$

While not typically an issue for constrained length data, such as Twitter tweets' 280 character limit, longer text sequences tend to bias extracted terms as they appear more frequently in a longer document. For instance, we target the variable word length forum Stormfront, whereby the de Gibert et al. [23] dataset includes sentence-long replies to rants and multi-paragraph miniature-manifestos for longer topical posts. Hence, we apply L2 normalisation to constrain the dataset's values to be between 0 and 1 (also referred to as the least squares approach), resulting in the following equation:

$$\hat{v} = \frac{\overrightarrow{v}}{\|\overrightarrow{v}\|}$$

We also consider grouping related words by topic and hate class category, which we group via cosine similarity using the TF-IDF vectors via the equation:

$$cos(\theta) = \frac{v \cdot w}{\|v\|\|w\|} = \frac{\sum_{i=1}^{n} v_i w_i}{\sqrt{\sum_{i=1}^{n} v_i^2}\sqrt{\sum_{i=1}^{n} w_i^2}} \tag{2.2}$$

TF-IDF clustering with cosine similarity equation

Whereby v and w represents two different document-to-document TF-IDF vectors. If the vectors are identical, then the words and frequency are identical—which is useful for plagiarism detection as a 1.0 cosine similarity denotes identical text sequences. Conversely, a cosine similarity of 0 identifies two different sequences of words without any shared terms.

As common words such as *the*, *what*, *to* frequently appear in any text regardless of topical content, they are marginalised in the TF-IDF calculation as these ‘stop words’ have both a high term frequency (TF) *and* high frequency in the overall collection (IDF), thus their TF-IDF score is low.

2.2.5 ERH Context Mining—an emerging academic area

We stratify the full ecosystem to developing ethical and representative hate speech detection via unifying sociolinguistics, computer science, ethics and law into the field we frame as ERH Context Mining. Context mining begins with social science analysis of what defines radicalisation, extremism, and hate speech as concepts. Drawing from our systematic literature review, we identified an international non-partisan consensus for each ERH definition—-without giving specific politically aligned examples [39]. We aggregate these observed definitions with the following modernised definitions for *Extremism*, *Radicalisation*, and *Hate speech* in the context of computational social media analysis and understanding online context:

Def. 1: Morphological Mapping & Consensus-building (Extrem.)

The congregation of users into collective identities (“in-groups”) in support of manifestly unlawful actions or ideas.

Def. 2: Ideological Isomorphism (Radicalisation)

The temporal movement of one’s belief space and network of interactions from a point of normalcy towards an extremist belief space. It is an approach to detecting radicalisation with an emphasis on non-hateful sentiment as ringleaders and/or influencers pull and absorb others towards their hateful group’s identity, relationships, and beliefs.

Def. 3: Outwards Dissemination (Hate Speech)

Targeted, harassing, or violence-inducing speech towards other members or groups based on protected characteristics.

Upon generating ERH annotation criteria, researchers begin the data mining element of context mining via the data selection, collection, and extraction elements displayed in Figure 2.5. Whereafter, the process shifts from social-science oriented criteria and dataset collection to ERH detection strategies utilising artificial intelligence. Responsible and Explainable Artificial Intelligence (XAI) requires harmonising social and computer science fields, as well as human-driven review processes to ensure that ERH detection or generation modules mitigate biases and do not infringe on user privacy. Elements such as identifying controversial platforms, synthetic text generation, and topic mapping to informative sources are all key research gaps identified in our systematic literature review and considered in this project's architecture. We further visualise the pipeline of ERH Context Mining, and its current research gaps, in Figure 2.5.

On an interdisciplinary level, Figure 2.4 displays the research recommendations and life cycle for software engineering with ERH systems. ERH Context Mining research and development should seek to balance the control and interaction between researchers; industry, such as social media platforms themselves; self-regulation, and law via government oversight—particularly for ethics and data sovereignty. Open-source intelligence (OSINT), which is information from publicly accessible data sources, is key to researching and deploying ERH detection models that protect user privacy. Likewise, ensuring self-regulation, such as restricting Prompt-GAN's access to academics upon their request and our review (i.e., semi-closed access), restricts our model from misuse and abuse by malicious actors-—essential given our Prompt-GAN architecture's ability to generate realistic and politically extreme hate speech.

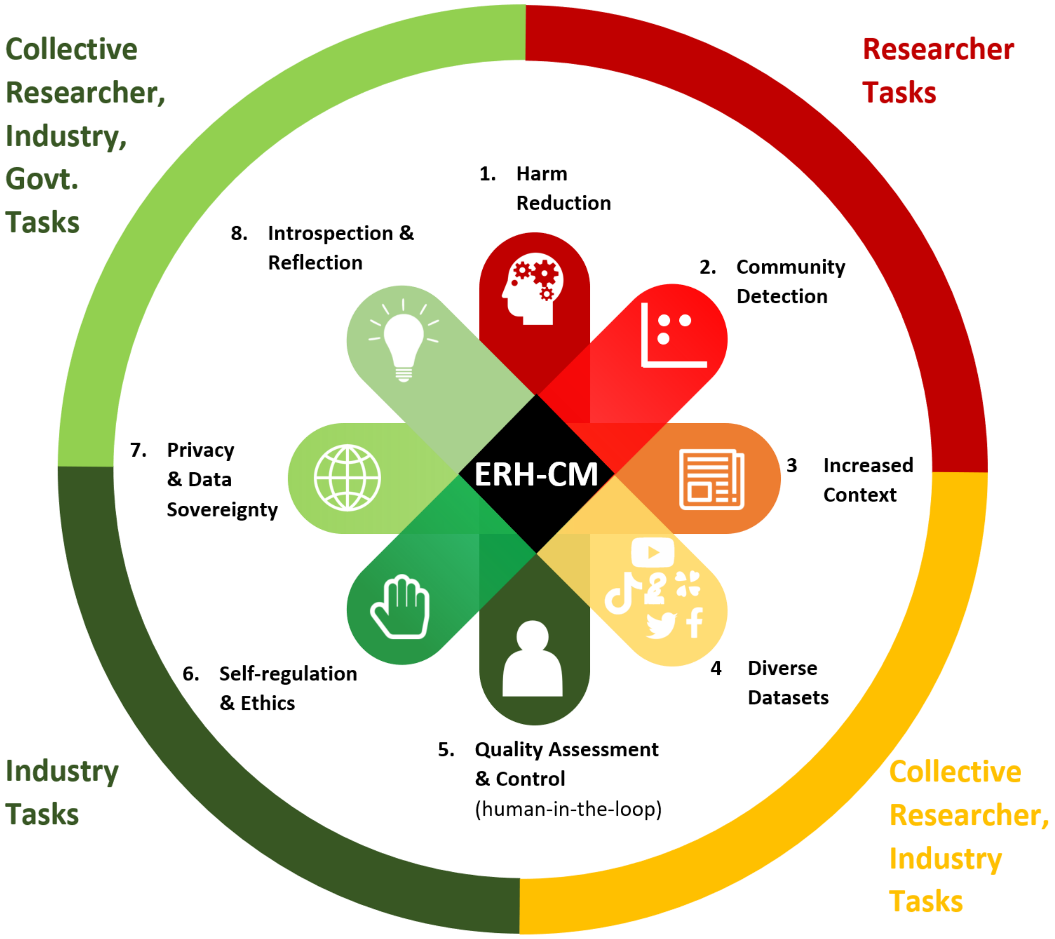

Collective Researcher, Industry, Govt. Tasks
Researcher Tasks
1. Harm Reduction
8. Introspection & Reflection
2. Community Detection
7. Privacy & Data Sovereignty
ERH-CM
3 Increased Context
6. Self-regulation & Ethics
4 Diverse Datasets
5. Quality Assessment & Control (human-in-the-loop)
Industry Tasks
Collective Researcher, Industry Tasks

Figure 2.4: ERH Context Mining (ERH-CM) eight core components for Research, Industry, and Government.

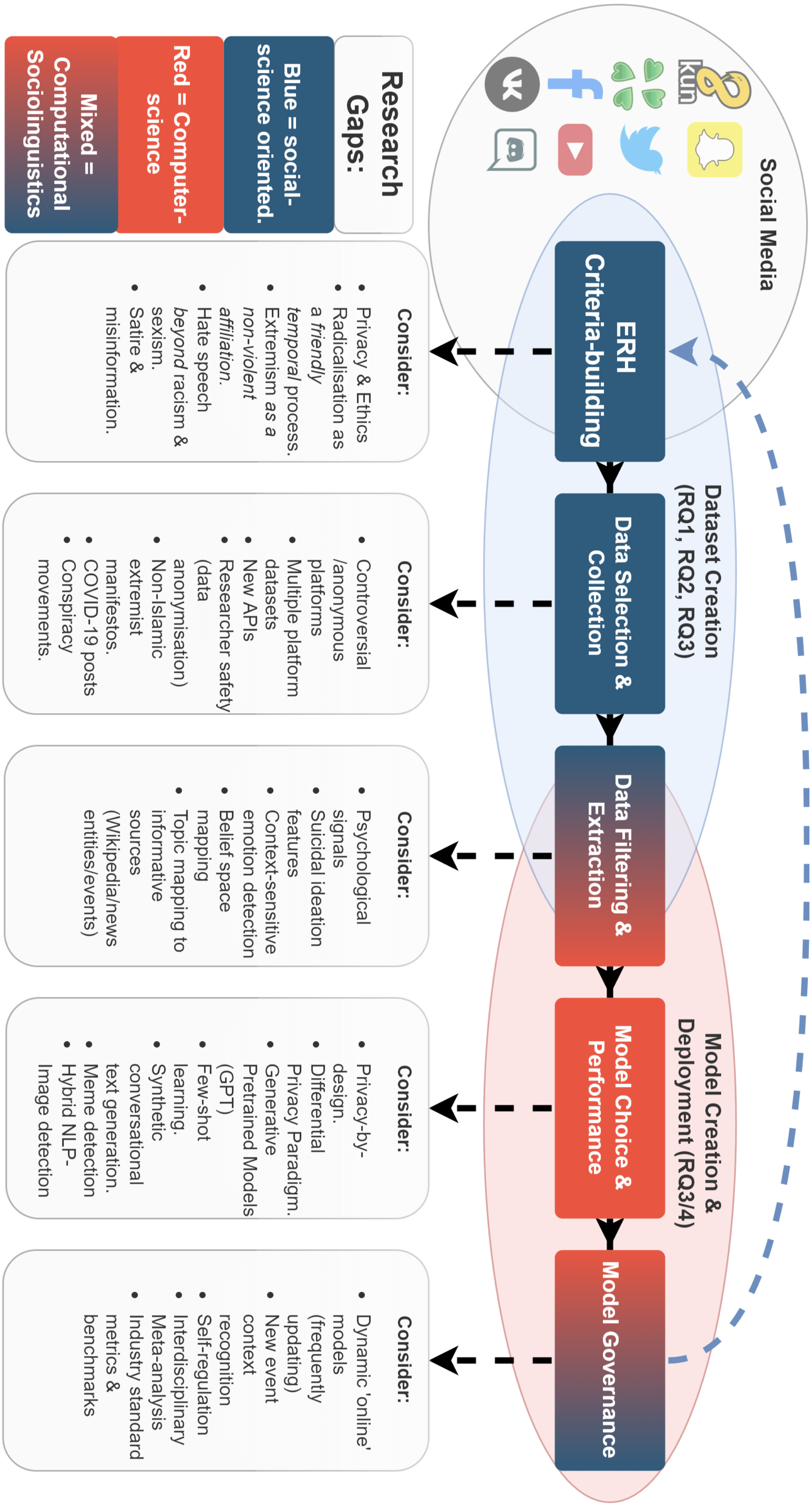

Social Media
Dataset Creation (RQ1, RQ2, RQ3)
Model Creation & Deployment (RQ3/4)
ERH Criteria-building
Data Selection & Collection
Data Filtering & Extraction
Model Choice & Performance
Model Governance
Research Gaps:
Blue = social-science oriented.
Red = Computer-science
Mixed = Computational Sociolinguistics
Consider:
• Privacy & Ethics
• Radicalisation as a friendly temporal process.
• Extremism as a non-violent affiliation.
• Hate speech beyond racism & sexism.
• Satire & misinformation.
Consider:
• Controversial /anonymous platforms
• Multiple platform datasets
• New APIs
• Researcher safety (data anonymisation)
• Non-Islamic extremist manifestos.
• COVID-19 posts
• Conspiracy movements.
Consider:
• Psychological signals
• Suicidal ideation features
• Context-sensitive emotion detection
• Belief space mapping
• Topic mapping to informative sources (Wikipedia/news entities/events)
Consider:
• Privacy-by-design.
• Differential Privacy Paradigm.
• Generative Pretrained Models (GPT)
• Few-shot learning.
• Synthetic conversational text generation.
• Meme detection.
• Hybrid NLP-Image detection
Consider:
• Dynamic 'online' models (frequently updating)
• New event context recognition
• Self-regulation
• Interdisciplinary Meta-analysis
• Industry standard metrics & benchmarks

Figure 2.5: ERH Context Mining pipeline—with key identified research gaps.

Chapter 3

Literature Review

This literature review chapter focuses on the sections and findings pertinent to Prompt-GAN's architectural design. For a holistic understanding of existing *multi-media ERH detection*, and more on *ERH Context Mining*, please refer to our full interdisciplinary Systematic Literature Review (SLR) [39].

Our SLR investigated the state-of-the-art approaches, datasets, socio-legal, and technical implementations used for *Extremism*, *Radicalisation*, and politicised *Hate speech* (ERH) detection. Unlike prior work, a computational approach to extremism includes political affiliation studies if the discussions contain hate speech. In context with the concept of Social Media Intelligence (SOCMINT), being the investigation of social media to understand user habits and beliefs, we frame *social media* data as any online medium where users can interactively communicate, exchange or influence others. We accepted external data sources, such as manifesto or news sites if they included interactive online sections—such as via comment sections. To preserve user privacy, we only consider information from publicly accessible forums and websites—in line with OSINT principles. We identified 51 studies from 2015-2021—a period selected to reflect an updated presentation of the state-of-the-art ERH detection methods compared to older reviews. No prior SLRs simultaneously considered *Extremism* (affiliation), *Radicalisation* (movement towards extremism ideology), and *Hate speech*. Furthermore, we present the first cross-examination of multi-media, community detection (i.e., relationships and networks between extremists), and Natural Language Processing (textual) detection of ERH in social media. We visualise each of the variations between labelled “supervised” individual hate speech and/or extremist posts, as well as the unlabelled automatic clustering (i.e., ‘unsupervised’) of online hateful and/or extremist groups in Figure 3.2. We targeted only peer-reviewed studies which investigated social media data via binary, multi-class, clustering, or score-based algorithms. Our iterative process for study selection included a title and abstract screen, which evaluated each study’s title and abstract for its compatibility with our search criteria and scoring regime [39]. Thereafter, we screened the full document for the Full Text Screening, and randomly sampled new studies to evaluate from the sampled studies bibliographies—through a process known as *snowball sampling.* Overall, we identified 51 studies to discuss throughout our SLR, with each study count outlined in Table 3.1, and the overall screening process visualised in Figure 3.1.

We also conduct a secondary esoteric literature review targeting existing prompt-engineering and synthetic text generation approaches, which we provide in the latter half of this chapter.

We conclude this Chapter in Subsection 3.2.2 with the key gaps in related work, and how we incorporate our proposed solutions within Prompt-GAN.

Table 3.1: Studies found and filtered

Screen Type	Study Count
Search Strings	251
Title and Abstract Screen	57
Full Text Screening	42
After Snowball Sampling	51

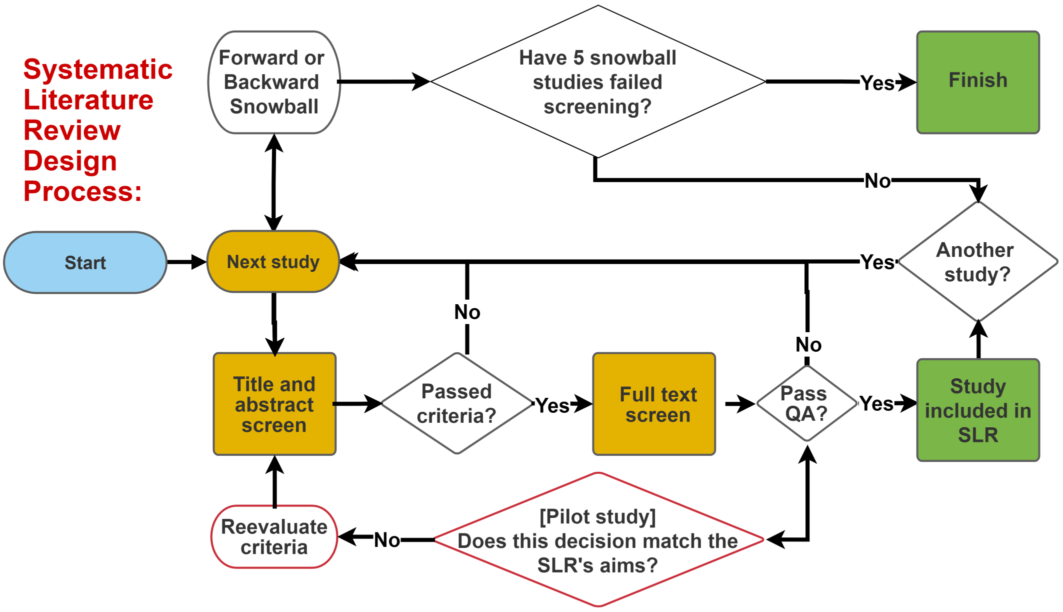

Systematic Literature Review Design Process:
Start
Next study
Forward or Backward Snowball
Have 5 snowball studies failed screening?
Yes
Finish
No
Another study?
Yes
No
Title and abstract screen
Passed criteria?
Yes
Full text screen
Pass QA?
No
Yes
Study included in SLR
Reevaluate criteria
No
[Pilot study] Does this decision match the SLR's aims?

Figure 3.1: The full protocol conducted for this Systematic Literature Review.

3.0.0.1 Threats to Validity

While we consider a concerted range of search strings, we recognise that ERH concepts are a wide spectrum. To focus on manifestly hateful, politicised, and violent datasets/studies, we excluded cyber-bullying or sentiment-detection studies. The potential overlap and alternate terms for ERH (e.g., sexism as “misogyny classification” [20]) could evade our search strings. Nonetheless, our pilot study, subsequent tweaks to our search method, and snowball sampling strategy minimises this lost paper dilemma.

This study does not involve external funding, and we declare no conflicts of interest.

3.1 Hate Speech Datasets—the Existing Benchmarks

We define a benchmark dataset as any dataset evaluated by three or more studies. The majority of studies used custom web-scrapped datasets or Tweets (51%) pulled via the Twitter API.

Hate speech datasets suffer significant researcher selection bias, with no

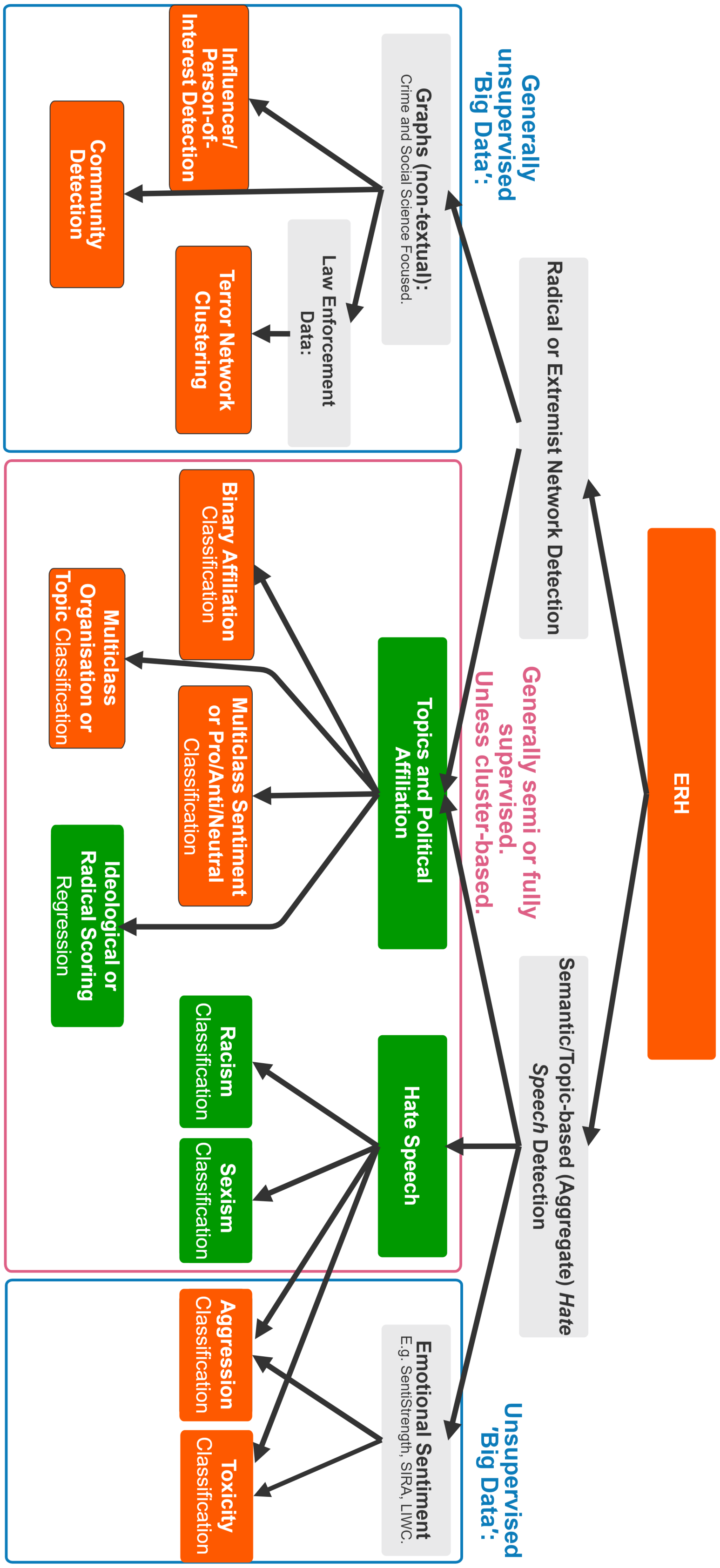

ERH
Radical or Extremist Network Detection
Semantic/Topic-based (Aggregate) *Hate Speech* Detection
Generally unsupervised 'Big Data':
Graphs (non-textual):
Crime and Social Science Focused.
Law Enforcement Data:
Influencer/ Person-of-Interest Detection
Community Detection
Terror Network Clustering
Generally semi or fully supervised. Unless cluster-based.
Topics and Political Affiliation
Hate Speech
Binary Affiliation Classification
Multiclass Organisation or Topic Classification
Multiclass Sentiment or Pro/Anti/Neutral Classification
Ideological or Radical Scoring Regression
Racism Classification
Sexism Classification
Unsupervised 'Big Data':
Emotional Sentiment
E.g. SentiStrength, SIRA, LIWC.
Aggression Classification
Toxicity Classification

Figure 3.2: Types of Extremism, Radicalisation, and Hate speech (ERH) research avenues. Hate speech and topical affiliation (white supremacy) targeted in this Prompt-GAN study.

Table 3.2: Datasets used by more than one study.

Dataset	Year	Categories	Platform of origin	Collection strategy	Used By
Waseem and Hovy [104]	2016	16914 tweets: 3383 *Sexist*, 1972 *Racist*, 11559 *Neutral*	Twitter	11-point Hate Speech Criteria	[49, 6, 104, 107, 77, 71, 72, 63, 48, 46]
FifthTribe [33]	2016	17350 *pro-ISIS* tweets	Twitter	Annotated pro-ISIS accounts	[73, 5, 82, 88]
de Gibert [23]	2018	1196 *Hate*, 9507 *Non-hate*, 74 *Skip* (other) post segments	Stormfront	3 annotators considering prior posts as context	[63, 48, 107, 23]
OffenseEval (OLID) [110]	2019	14100 tweets. (30%) *Offensive* or Not; *Targeted* or *Untargeted* insult; towards an *Individual*, *Group*, or *Other*	Twitter	Three-level hierarchical schema, by 6 annotators	[110, 113, 48, 59]
HatEval [8]	2019	10000 tweets distributed with *Hateful* or Not, *Aggressive* or Not, *Individual* targeted or *Generic*	Twitter	Crowdsourced via Figure Eight, with 3 judgements per Tweet	[63, 103, 104, 107, 48, 8, 76]
Davidson et al. [21]	2019	25000 tweets: *Hate speech*, *Offensive*, *Neither*	Twitter	3 or more CrowdFlower annotators per tweet	[21, 63, 48, 107, 46, 72, 71]
TRAC [53]	2018	15000 English and Hindi posts; Overtly, Covertly, or Not Aggressive	Facebook	Kumar et al. [54] subset, 3 annotators per post, comment or unit of discourse	[49, 48, 63]

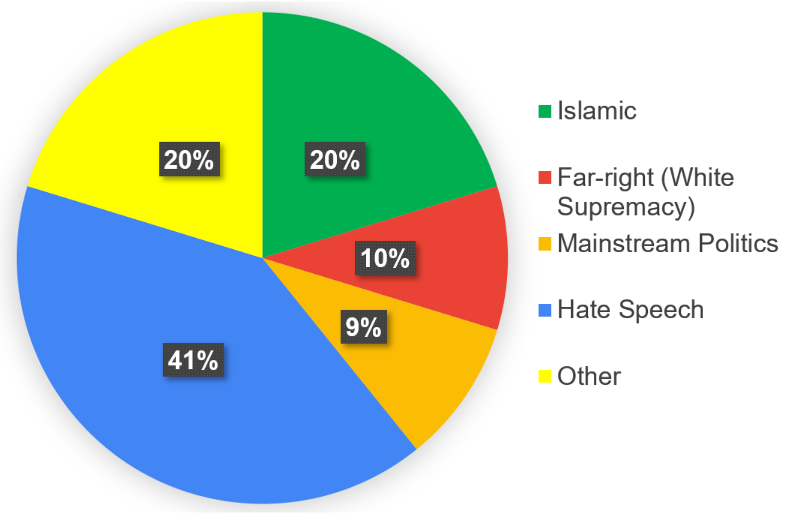


Figure 3.3: Distribution of classification target across the 51 studies.

studies utilising data or groups from Oceanic countries due to the global skewed focus on the US and the Middle East in ERH research. Specifically, despite the decline of the Islamic State as a conventional state actor post-2016, 31.6% of US-originating studies targeted Islamic extremism, compared to 15.8% focusing on violent far-right groups. Despite more Islamic extremist studies from US-oriented research, over 90% of terrorist attacks and plots in the US were from far-right extremists in 2020. Across all studies globally, far-right white supremacy only constitutes 10% of studies as displayed in Figure 3.3

Waseem and Hovy offer a quintessential benchmark due to its frequent use by 10 studies, with its comprehensive 11-point hate speech criteria [104]—unique given that only 20% of the 51 studies offered a legal or social definition or criteria for annotating hate speech. Unfortunately, revised 2021 Twitter Academic API regulations removed the ability to pull from suspended accounts—removing the ability for researchers to use datasets requiring the API [9]. Nonetheless, alternate Twitter-based benchmark datasets such as the Davidson et al. Hatebase-Twitter dataset includes archived Tweets stored on Git, thus remaining accessible to researchers [91].

Thus, we select three datasets based on our literature review to test Prompt-GAN's ability to generate synthetic hate speech and conduct transfer learning to prove its 'generalisation' capability. We define 'generalisation' as the ability to train Prompt-GAN on one dataset's training data ('dataset A'), then generate synthetic hate speech but instead test this synthetic data on the real data from the external unseen 'datasets B and C'. The concept of testing another dataset imbues the concept of *transfer learning* and will demonstrate Prompt-GAN's ability to create multi-class, multi-topic and multi-platform hate speech.

The three datasets we train Prompt-GAN on are:

1. **de Gibert et al. (DG) Stormfront dataset** [23]—to highlight and address the underrepresented field of far-right extremism and hate speech research—and its impact on New Zealand's security as evident in the 2019 Christchurch shootings. The Southern Poverty Law Center describes Stormfront as "the first major hate site on the Internet... [whereby] the site has been a very popular online forum for white nationalists and other racial extremists" [94]. The data is of variable length, ranging from small replies to long diatribes such as political statements and calls to action which constitute a form of 'mini-manifesto'. This dataset will help demonstrate Prompt-GAN's ability to generate long human-like,

contiguous, and class-relevant hate or non-hate posts. Non-hate data are from Stormfront posts that are not explicitly hateful. The longest post is 2153 characters, and the median post length is 153 characters.

2. **Davidson et al. (Hatebase-Twitter) dataset** [21]—a benchmark dataset used by seven datasets observed in our SLR. We test Prompt-GAN's ability to generate restricted length Tweets—which are up to 280 characters long. The multi-class 'Hate', 'Offensive' and 'Neither' class labels will discern the difference between general offensive *hatred*, compared to targeted *discrimination* of protected characteristics à la *hate speech*. The median post length is 81 characters.

3. **Implicit-Explicit hate speech** [28]—contains Twitter tweets as labelled as either *Explicit hate*, *Implicit hate*, or *Non-hate*. Explicit hate denotes verbal attacks which are "direct and leverages specific keywords" and is the default hate definition for binary datasets. Implicit hate is a novel category containing "coded or indirect language that disparages a person or group on the basis of protected characteristics like race" [28, p. 345] used to evade bans via plausible deniability. Implicit hate also includes latent linguistic features such as sarcasm for a discriminatory effect, coded language, references, dehumanising stereotypes and motivated disinformation. Implied hate often relies on deliberate logical fallacies to undermine another group, while explicit hate is direct, targeted and threatening. The non-hate class acts as the control in this three-class non-overlapping dataset. The median post length is 89 characters.

Furthermore, the variety in platforms (Stormfront posts and Twitter tweets) and ideological affiliation will help identify biases in language models based on the topics, individuals, and politics emanated from GPT-2/Neo in the synthetic posts. The selection of datasets offers posts from a white supremacist extremist affiliation, as well as generic racist, sexist and other discriminatory posts via the Twitter datasets.

3.2 Hate speech Discriminators and Generators—Observing the State-of-the-Art

For textual NLP studies, researchers tended to classify hate speech by converting the input into word embeddings via Word2Vec, GloVe, or frequent words via Term Frequency-Inverse Document Frequency (TF-IDF); then parsing it into Support Vector Machines, decision trees, or logistic regression models [39]. As these embeddings do not account for word order, context and nuance are often lost—leading to higher false positives on controversial political threads. Conversely, another approach to hate speech classification is through deep learning, which implements context-sensitivity through positional and contextual word embeddings for higher classification performance. Deep learning approaches such as Bidirectional Encoder Representations from Transformers (BERT), Convolutional Neural Networks (CNN), and Bidirectional attention Long Short-Term Memory (BiLSTM) deep learning approaches frequently outperformed non-deep machine learning approaches, as evident in Table 3.3.

Overall, the high F1-score performance of state-of-the-art BERT, BiLSTM, and CNN-GRU models represent a recent trend towards deep learning approaches for hate speech discrimination—with Figure 3.4 displaying the shift of model choice over time due to these advances in deep learning models.

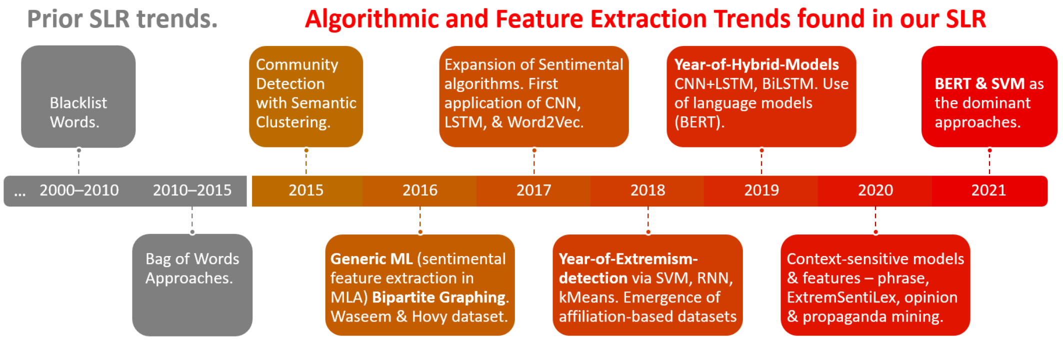

Prior SLR trends.
Algorithmic and Feature Extraction Trends found in our SLR
Blacklist Words.
Community Detection with Semantic Clustering.
Expansion of Sentimental algorithms. First application of CNN, LSTM, & Word2Vec.
Year-of-Hybrid-Models CNN+LSTM, BiLSTM. Use of language models (BERT).
BERT & SVM as the dominant approaches.
... 2000–2010
2010–2015
2015
2016
2017
2018
2019
2020
2021
Bag of Words Approaches.
Generic ML (sentimental feature extraction in MLA) Bipartite Graphing. Waseem & Hovy dataset.
Year-of-Extremism-detection via SVM, RNN, kMeans. Emergence of affiliation-based datasets
Context-sensitive models & features – phrase, ExtremSentiLex, opinion & propaganda mining.

Figure 3.4: Patterns of adoption for ERH detection algorithms over time. Colour change ordered by F1-score trend (low to high). Brown = 0.75 F1-score on benchmark datasets, Red = 0.9 F1-score, Grey = No Data.

Hence, for the discriminator component of our Generative Adversarial Network approach, we consider variants and derivatives of BERT models due to their top three F1-score performance on the Davidson et al. [21] Twitter, and de Gibert et al. [23] Stormfront datasets. The specific approaches and code for the SP-MTL LSTM (highest performing Stormfront hate classification model) and the Davidson et al. BiLSTM model are not provided—thus we utilise BERT models outlined in Section 4.2.3. While BERT, attention-layered Bidirectional Long Short-Term Memory (BiLSTM), and other ensemble DLAs attain the highest F1-scores, no studies consider their performance trade-offs with their high computational complexity. Thus, we consider computational performance in our design by analysing the RAM and Video-RAM requirements, and the model's runtime complexity. Furthermore, we act on our full SLR's [39] recommendations for further research in prompt-engineering (via our text generation module), and distilled classification models to reduce runtime complexity—leading to our selection of DistilRoBERTa per our discriminator tests outlined in Section 5.1.2.

We target text-based generation and neural network approaches for detecting hate speech, as non-textual community detection (i.e., relationships such as follower/following networks) and traditional non-deep machine learning studies resulted in lower classification F1-score by ~0.15 and ~0.2 respectively.

Table 3.3: Models ranked by F1-score for the benchmark datasets across studies (inter-study evaluation).

Dataset	1st Highest	2nd Highest	3rd Highest
Waseem and Hovy [104]	0.966 (BERT with GPT-2 fine-tuned dataset [107])	0.932 (Ensemble RNN [77])	0.930 (LSTM + Random Embedding + GBDT [6])
FifthTribe [33]	1.0 (RF [73])	0.991-0.862 (SVM [5])	0.87 (SVM [82])
de Gibert et al. Stormfront [23]	0.859 (SP-MTL LSTM, CNN and GRU Ensemble [48])	0.82 (BERT [63])	0.73 (LSTM baseline metric [23])
TRAC FB [53]	0.695 (CNN + GRU [48])	0.64 (LSTM [53])	0.548 (FEDA SVM [49])
Davidson et al. Twitter [21]	0.923 (BiLSTM with Attention modelling [72])	0.92 (BERTbase+CNN / BiLSTM [71], 0.86 with debias module)	0.912 (Neural Ensemble [63])
HatEval [8]	0.7481 (Neural Ensemble [63])	0.738 (LSTM-ELMo+BoW) [76]	0.695 (BERT with GPT-2 fine-tuned dataset [107])
OffensEval [110]	0.924 (SP-MTL CNN [48])	0.839 (BERT [113])	0.829 (BERT 3-epochs [59])

3.2.1 Existing synthetic data generation methods

We identified only two studies that considered synthetic dataset generation, namely the LSTM-CNN HateGAN by Cao and Lee [16], and fine-tuning GPT-2 for the MegaSpeech corpus by Wullach et al. [107] However, no studies consider prompt-engineering techniques on pre-trained language models, nor utilising pre-trained models for inference-tasks without expensive and resource-heavy fine-tuning. Both Cao and Lee [16], and Wullach et al. [107] also frame annotation time, the financial cost of crowdfunded data annotation, and human biases as the key motivations for their approaches. However, Wullach et al. only consider financial cost, *without accounting for* memory or runtime cost—which we consider in our lower memory inference-only prompt-tuning approach. Thus, our approach enables Prompt-GAN to use larger models like GPT-2-XL and GPT-Neo-2.7B on our commodity desktop setup outlined in Section 5.1.

The fine-tuned GPT-2 approach by Wullach et al. [107] utilised the benchmark datasets from Waseem and Hovy [104], Davidson et al. tweet-based dataset [21]; de Gibert et al. Stormfront dataset [23]; and other datasets from Founta et al. [35] and SemEval [8]. Specifically, Wullach et al. merged these datasets to fine-tune GPT-2-large (764M parameters) to create synthetic hate and non-hate sequences, later selecting the top 100k sequences with the highest respective hate or non-hate class probabilities from a pre-trained BERT classifier trained on the original real corpora [107]. However, the capability for generalisation in multi-task learning on a fine-tuned dataset presents a unique challenge vis-à-vis the catastrophic forgetting dilemma. Tests training a BERT classifier on one dataset and testing on another dataset altogether (i.e., transfer and multi-task learning), resulted in a 6.95% reduction in test accuracy compared to a model using additional fake MegaSpeech data. However, Wullach et al. do not specify their memory usage or CPU/GPU setup, nor the train time and complexity for fine-tuning GPT-2 on the 89,514 annotated posts [107].

The second approach observed for synthetic dataset generation was the HateGAN LSTM-CNN approach by Cao and Lee [16]—utilising an LSTM-

CNN text sequence generator, with a pre-trained toxicity scoring discriminator. The value of the pre-trained toxicity scorer determined the alterations to the weights during the backpropagation pass on the LSTM-CNN model. Like the GPT-2 fine-tuned model, Cao and Lee also utilised the benchmark Davidson et al. and Waseem and Hovy Twitter datasets. For the Davidson et al. dataset, their HateGAN CNN-LSTM model attained a 0.894 F1-score using the synthetic text [16], while the GPT-2 augmented (synthetic and real mixed data) data from Wullach et al. attained a 0.865 F1-score [107].

3.2.2 Key gaps addressed in Prompt-GAN's design

Given the automatic or fast suspension of hate speech on conventional platforms such as Twitter, collecting hate speech data can be a costly, timely, and ethically risky endeavour. Collecting hate speech can be difficult for human discrimination—often requiring group consensus through paid annotation platforms such as Amazon Mechanical Turk or Figure Eight [6, 8, 21, 35, 37, 63, 71, 106, 107]. Extracted posts online may also lack a balance between classes—particularly as moderated platforms enforce community guidelines and remove illicit posts. Thus, paid annotation can result in annotated datasets that contain an insufficient quantity of hate speech—whereby Waseem and Hovy observed that deep learning classifier performance stabilises with at least 1000 instances per class [104]. For instance, Waseem and Hovy's dataset required processing 136,052 tweets and annotating 16,914 tweets. Twitter's 2021 decision to restrict access to suspended account tweets via the Academic API also invalidates future researchers using datasets where the tweets are not stored in a repository. Furthermore, constantly identifying and extracting real data from extremist forums also raises ethical risks and risks to a researcher's safety, due to an extremist's persecution complex—the irrational belief that they're being targeted by a foreign researcher group who is 'out to find or get them'. Hence, synthetic data generation enables researchers to investigate and replicate the patterns of hate, rather than specific quotes from real individuals*.

We specifically address the bias towards Twitter and Middle Eastern data observed in our SLR [39] by including the far-right white supremacist forum Stormfront—with posts annotated by de Gibert et al. [23]. Moreover, we address the lack of multi-platform-oriented hate speech studies by cross-examining GPT-2/Neo's ability to create synthetic long-text Stormfront posts, and up to 280-character Twitter tweets. Given that hate speech is often politically diverse and nuanced, we also consider multi-class labels—specifically explicit hate speech (*direct and targeted with dehumanising intent*) and implicit hate speech (*latent references and logical fallacies to undermine an individual or group based on their protected characteristics*). We also consider *ideological alignment* through the white supremacy element of Stormfront, as well as separating offensive speech from hate speech via the Davidson et al. Twitter dataset [21]. Hence, the investigation on latent hate, explicit hate, offensiveness, and politically aligned hate across multiple platforms will demonstrate Prompt-GAN's adaptability to multiple topics and domains—and display the radicalisation risks of pre-trained commercially-available GPT models.

Finally, we aim to computationally evaluate the risks and dangers to the baseline GPT-2/Neo model—as we demonstrate that GPT can produce realistic and dangerous hate/extremist speech without any modifications to the model itself (only by prompt-tuning). This risk analysis includes exploring the out-of-corpus topics discussed in the synthetic datasets to identify the biases and geopolitical knowledge of GPT-2/Neo.

* However, GPT is prone to replicating copyrighted work per its training corpora [38, 7]. In our experiments, we identified GPT-2/Neo's ability to replicate Twitter handles and names. Based on a reverse search, handles do not link to real tweets or posts. The only handles relating to real figures were those for public figures or entities such as @realdonaldtrump, @hillaryclinton, and organisations like @HuffingtonPost. However, names and potential handles generated could link to real non-public figure accounts by circumstance.

Chapter 4

Research Design

This chapter outlines the core research questions and architectural design to investigate whether we can create synthetic types of hate and non-hate speech with minimal overlap. We define and quantify this objective by constructing a series of metric-driven experiments to form a *digital Turing test*. This digital Turing test concept includes blind black-box testing with a third-party hate classifier, readability metrics, social topic analysis, comparing classification performance to real data with our discriminator, and whether the synthetic data can achieve or outperform the existing HateGAN (LSTM-CNN approach) and MegaSpeech corpus (GPT-2 fine-tuning) data via classification F1-score. In Section 4.1, we present the three research questions, with the first two covering the prototyping and validation of our prerequisite assumptions, validating our synthetic data compared to another corpus. Our third research question observes whether GPT-2/Neo produces generalisable posts which can discuss new topics, domains, and types of hate speech/extremism where competing architectures cannot due to a lack of pre-trained knowledge or from the catastrophic forgetting dilemma. Thereafter, we present the architectural pipeline and design for Prompt-GAN in Section 4.2, as well as our prerequisite manual prompt-tuning and embedding prototype to aid the final design for the full Prompt-GAN pipeline per Section 4.3.

We consider GPT-2-XL (1.5 billion parameters), and GPT-Neo (2.7 billion parameters), both via local inference on the machine specifications outlined in our experimental setup in Section 5.1. For brevity, we refer to these language models as the NLMs (Neural Language Models) in our experiments.

For Prompt-GAN to work, we hypothesise that we can alter the belief space of NLMs via automated prepended tokens such that:

1. NLMs pre-trained corpus contains and understands hate speech to the extent that it can replicate it.
2. That OpenAI and EleutherAI (GPT-Neo) have all failed to debias and detoxify their NLMs. If this assumption holds, this raises significant ethical concerns regarding NLMs viability for commercial use and the risk of automated hate bots by state or non-state actors to destabilise society and undermine mental health.
3. NLMs can understand instruction prompts and create on-topic/contiguous text across multiple sentences, including for smaller local models.
4. That we can optimise prompts through an automated algorithm.
5. That an NLM can discern non-binary hate classes (i.e., implicit and explicit hate), offensiveness which is not hate speech, as well as ideological affiliation to create *politicised* extremist hate speech.

4.1 Research Questions

Our three research questions target whether NLMs can be *radicalised* to produce specific desired tokens—with a focus on driving the NLM from abstract sequences to *hateful* and *ideologically affiliated* sequences. Furthermore, we consider whether the output from the state-of-the-art language models can mimic online speech as defined by our evaluation strategy. Finally, we consider whether the extensive pre-trained corpus, ~40GB for GPT-2 and ~886GB for GPT-Neo, improves transfer learning tasks by using its pre-trained knowledge to create contextualised and on-topic synthetic posts when presented with an out-of-corpus topic or social media platform to mimic.

Thus, our three research questions are:

RQ1 ***Can neural language models produce topic and platform-specific hate speech with competitive F1-scores and toxicity metrics compared to a real hate speech corpus?***

RQ2 ***Can our model generalise to other datasets via transfer learning?***

RQ3 ***Can our model create synthetic online posts which target topics and other group affiliations outside of the training datasets?***

For RQ1, we define a competitive synthetic dataset whereby training a hate speech discriminator on mixed (50% real data, 50% fake) or all synthetic data (100% fake data for each class) results in classification F1-scores on real data which is within a 5% F1-score delta to an all-real training dataset. In our experimental setup, we consider multiple scenarios consisting of using synthetic data to supplement/"boost" a real-corpus dataset by using a fixed proportion of real data and supplementing the training data with synthetic posts to make a larger overall dataset. Furthermore, we test an augmented "mixed" experiment setup using a fixed total amount of posts but with a proportion of the synthetic data added to the real training data. Finally, we test the synthetic data as the training dataset in our *replacement* tests. In all scenarios, we cross-examine test macro F1-scores on a test dataset consisting of only real data.

RQ2 extends RQ1 by training with the synthetic data and evaluating classification performance on the test data from different datasets. Hence, we consider Prompt-GAN's flexibility to create out-of-corpus topics and domains—such as testing real Stormfront posts for a model trained on our synthetic Twitter hate speech corpus. We consider our model "generalisable" when the test real data's classification F1-scores matches or exceeds the baseline real data only model. The degree of generalisability will vary depending on the performance of Prompt-GAN, as it may produce lower F1-scores in certain scenarios or datasets. Hence, we consider this RQ successful if the supplemental or mixed real-synth training data scenarios outperforms an all-real training data classifier. We also investigate an external hate score classifier to compare the mean hate probability between the synthetic and real data—of which neither the real or fake data will be a part of the third-party model's training data. We consider the third-party pre-trained BERTweet hate speech detection classifier by Pérez et al. [78], trained on hateful tweets from the Basile et al. HatEval dataset [8]. The Basile et al. dataset contains annotated tweets "mainly collected in the time span from July to September 2018" [8, p. 55]; while the ElSherief et al. dataset collected tweets from January 1, 2015, and December 31, 2017; and the Davidson et al. dataset collected tweets from at least prior to its publication in 2017. Hence, the third-party 'blind' BERTweet hate classifier will not have overlapping tweets from our three targeted datasets—thus not biasing our results. Furthermore, cross-examining the ElSherief et al. and Davidson et al. datasets together did not indicate evidence for duplicate tweets from the same authors. For comparisons between NLM's we cross-examine the mean hate probability on the blind third-party BERTweet classifier, the macro F1 score on the real-world test data from our trained discriminators, and a linguistic analysis of the topics and readability of the synthetic text.

For RQ3, this involves cross-examining the topics discussed in the synthetic text and ensuring it covers out-of-corpus entities and events. In essence, this involves our concept of extracting related topics via BERTopic, GPT-2/Neo's latent understanding of history and geopolitics via its pre-trained corpus, and Wikipedia articles. We investigate our domain expansion strategy through linguistic analysis, as well as F1-score metrics for transfer learning. We also provide examples of synthetic hate speech on out-of-corpus topics. Thus, we demonstrate Prompt-GAN's adaptability and capability to generate diverse multi-topic and internationalised datasets.

4.2 Architectural Design

In this section, we outline the three modules that form the Prompt-GAN architecture consisting of the prompt and vocabulary generator, the GPT-2/Neo text generation module, and the discriminator network to feed back to the prompt-generator as a form of a policy engine. Figure 4.1 displays a visualised summary of all three components discussed in this section.

4.2.1 Architecture for the Prompt Generator

We do not consider using a traditional neural network for Prompt-GAN's prompt generator, as this would require considerable resources to train the prompt generator to understand the semantics and grammar of natural language *in addition* to searching and understanding how to optimally create an instruction-based prompt. Another neural network approach would be to attach a smaller *language model* where the input is a random latent space which, through the layers of the generator language model, outputs a numeric embedding-based prompt for the larger language model to generate the synthetic speech. However, fine-tuning a smaller language model for prompt generation incurs significant training resources as each training iteration requires fine-tuning and inference operations on the prompt-generating language model *and* inference on the larger language model. The viable alternates would be to either seed the prompt-generator language model with a starting instruction prompt (e.g., "Write a tweet: ") and perturb the embeddings of the prompt to find new words, topics, and phrases to achieve the desired hate speech output; or implement a textual non-neural network string-builder approach. We employ the latter approach of building a text-based prompt string as we have greater control to select and refine the text. Specifically, we can build a vocabulary of class-specific words to prepend as keywords/topics to a static pool of instruction prompts. Our model can then test variations of prepended words to observe the resulting NLM synthetic text and its impact on the discriminator's

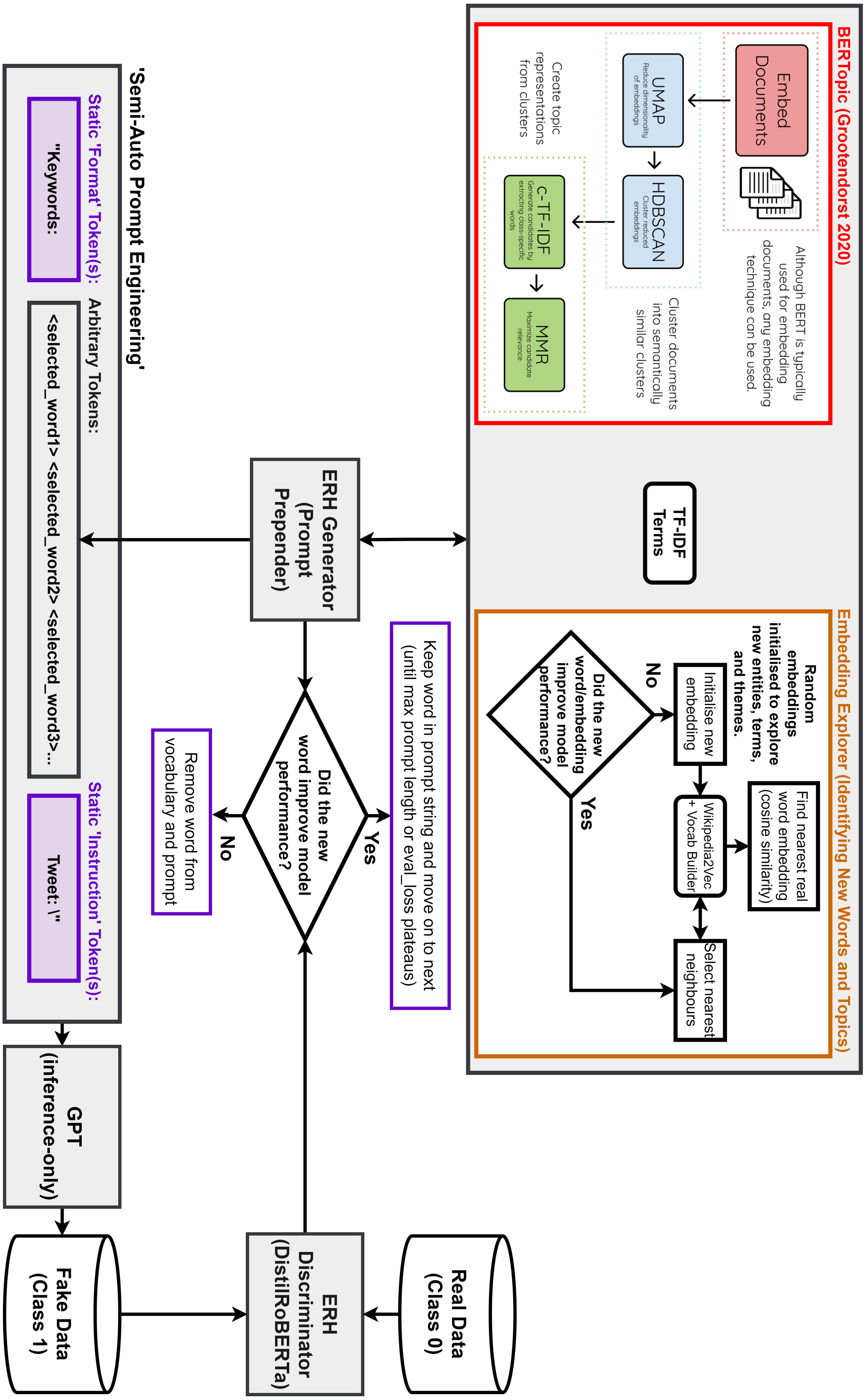

Prompt Token Vocabulary
BERTopic (Grootendorst 2020)
Embed Documents
Although BERT is typically used for embedding documents, any embedding technique can be used.
UMAP
Reduce dimensionality of embeddings
HDBSCAN
Cluster reduced embeddings
Cluster documents into semantically similar clusters
Create topic representations from clusters
c-TF-IDF
Generate candidates by extracting class-specific words
MMR
Maximize candidate relevance
TF-IDF Terms
Embedding Explorer (Identifying New Words and Topics)
Random embeddings initialised to explore new entities, terms, and themes.
Initialise new embedding
Find nearest real word embedding (cosine similarity)
Wikipedia2Vec + Vocab Builder
Select nearest neighbours
Did the new word/embedding improve model performance?
No
Yes
ERH Generator (Prompt Prepender)
Keep word in prompt string and move on to next (until max prompt length or eval_loss plateaus)
Did the new word improve model performance?
Yes
No
Remove word from vocabulary and prompt
'Semi-Auto Prompt Engineering'
Static 'Format' Token(s):
"Keywords:
Arbitrary Tokens:
<selected_word1> <selected_word2> <selected_word3>...
Static 'Instruction' Token(s):
Tweet: \"
GPT (inference-only)
Fake Data (Class 1)
ERH Discriminator (DistilRoBERTa)
Real Data (Class 0)

Figure 4.1: Architecture diagram for Prompt-GAN's Prompt Generator - NLM Text Generator - DistilRoBERTa Discriminator GAN pipeline.

loss. This approach is semi-context sensitive, as the word order of the prompt matters due to the trial-and-error approach of prepending words one-by-one from right-to-left of the input prompt. Throughout this subsection, we contextualise this prompt-building process, as well as our methodology for computationally searching for class-relevant words (singular) and entities/events (possibly multi-word, such as the ‘Christchurch mosque shootings’) to add to our vocabulary.

4.2.2 Building the prompt string—an issue of computational complexity and context-sensitivity

The first prerequisite component for the Prompt-GAN architecture is building a size-constrained vocabulary of words to sample and test by prepending them to the input prompt. The generator passes this input prompt with the sampled token(s) into the NLM for text generation to create a batch of synthetic textual posts, which we test in comparison to the real data via the discriminator model. Figure 4.1 presents this pipeline starting from the selection of a new word to prepend (or replace) to the prompt, generating a batch of fake online posts using that prompt, and evaluating its impact on the discriminator to determine whether the new word remains. Whereas Figure 4.2 displays the vocabulary-building architecture discussed in the following section.

As an example of how this prompt-builder approach works, consider the following simplified example for generating a synthetic tweet about cooking:

1. Firstly, we parse the real training corpus through TF-IDF, BERTopic, and Wikipedia2Vec to identify in-domain and out-of-domain topics, entities, and terms relevant to the task. Hence, the output from a tweet corpus might return a vocabulary of [chocolate, cake, ice cream, house, store]. Consider the static instruction as “Write a Tweet:”.

2. Secondly, the prompt-prepender approach randomly selects one of the terms, entities, or topics from the vocabulary to prepend to the static

instruction. If "cake" was the first sample, then the prompt would be "cake. Write a Tweet:".

3. If the discriminator loss increases as it creates more on-topic/realistic-to-the-dataset output, then we update the current prompt.

4. For a second sample, we randomly get 'house', for the new prompt string of "house cake. Write a Tweet:". If this decreases discriminator loss as it is not relevant, then we remove this word from the vocabulary and the string—placing us back to the second step. This process continues from 2-4 iteratively until three consecutive failures to prepend or substitute a word, entity, topic, or term.

If more than three failures to prepend a new token occurs, we shift into substitution mode, whereby instead of prepending new words to the prompt, we substitute and test existing words in the prompt string, including the fixed static prompt and stop-words (i.e., words that form a sentence's structure but is not intrinsic to the topic, such as words like 'the', 'what', 'a', 'and'). For instance, if the prompt is "***cake*** ice cream chocolate. Write a Tweet: ", we randomly select a word and replace it, such as "***strawberry*** ice cream chocolate. Write a Tweet:". If the substitution increases discriminator loss, then the word remains. However, if three consecutive substitutions fail to generate more realistic text, then the overall training process ceases—thus forming our 'three-strikes rule'. We limit our training to three consecutive failed attempts to prepend and three consecutive failed attempts to substitute a word based on our optimisation tests to balance training time and realism.

Our training process can also cease after a fixed number of prepended words, or when the vocabulary depletes. We also consider substituting words from the instruction prompt to build an ideal instruction in the same process as the above. We build a bag of instructions and a bag of prepended tokens, which we then use to create a list of tuned prompts to create the multi-topic dataset. For our full synthetic dataset, we create one unique class-and-dataset-specific prompt per 2500 synthetic posts.

As we produce multiple synthetic online posts per prepended word to train the discriminator, the average-case training complexity of our model is:

$$O(\frac{n * l * (s * f)}{2})$$

- n being the number of words in the vocabulary.
- l being the max length of the prompt.
- s being the number of substitutions per failure.
- f being the max number of failures allowed.

N.B: based on the '3-strikes rule' used based on hyper-parameter tuning, $s = f$, therefore equivalent to s^2. The average-case assumes an equal probability for a prepended or substituted word to fail a training step. In practice, the prompt will stabilise after ~7-10 terms where the probability that a new token fails to change discriminator loss increases as discriminator loss plateaus.

4.2.2.1 Architecture for Building the Vocabulary

The fundamental design challenge is to efficiently reduce the 470,000 possible English words [68], and ~6,450,000 historically significant events/entities (as identified in our retrained Wikipedia2Vec model). Without reducing the size of our vocabulary to concepts/terms relevant to hate/non-hate classes, we would need to conduct at least 140 million inference operations which would take ~42 million seconds (~1.3 years) to train, given the ~0.3 seconds per synthetic post generation for the sample of 20-40 posts per prepended token on our system.

Hence, we build our vocabulary by identifying frequently used terms, topics (via BERTopic), entities and events (via Wikipedia) discussed within the corpora of the three hate speech datasets. While we test contextual embeddings and approaches vis-à-vis manipulating the input numeric embeddings for prompt-tuning, our preliminary analysis did not identify any patterns between classes (see Subsection 4.3.2 for visualisations). Hence, we only consider a token-based prompt-tuning approach. To target our belief space of hateful topics, politics, views, and discussion points—we consider the following textual approaches for building our vocabulary of relevant terms:

TF-IDF:

We split the data between classes and extract frequent words relating to different classes of hate speech using TF-IDF with L2 regularisation. L2 regularisation ensures that longer posts do not bias the selected terms due to their more likely appearance [105].

BERTopic:

Architecturally, our use of BERTopic selects relevant topics (e.g., countries, history, and related subtopics) both directly referenced in the corpus, as well as related concepts imbued within the BERTopic model.

The advantage of BERTopic in our vocabulary architecture is for identifying related topics and terms that may not appear within the Twitter and Stormfront corpora. For instance, the white supremacist Stormfront dataset revolves around white-on-black racism, while Twitter's hate contains generalised international discrimination including black-on-white racism. We can identify these aggregate trends to identify specific 'discussion points' to ensure that the NLM text generator remains on topic and creates realistic talking points. We exclude a random sample of topics for future prompt-generation sequences to enable diverse topical prompts. For instance, one prompt may focus on generating hate speech surrounding *racism* and *immigration*, while another will use shelved topics surrounding *sexism* and *gaming*.

Wikipedia2Vec:

While topics and subtopics are useful to identify the overall context of the discussion (e.g., "politics", "liberals", "Ireland", "Education", as observed from the Stormfront data), we further seek to simulate discourse around significant events and entities. While traditional hate speech datasets may target a specific timeframe or a forum topic, we seek to enable the ability to create synthetic text which discusses new out-of-corpus topics and events. For instance, fine-tuning the 2020-era GPT-Neo on a hate speech dataset collected from 2016 will not be able to generate synthetic hate speech supporting the 2019 Christchurch Shooter—as the knowledge of the attack will not be in the training dataset, and the original latent knowledge of the event in the pre-trained model will be lost due to fine-tuning's issue of catastrophic forgetting.

Hence, we sample the top topics identified from BERTopic to generate different discussion points across multiple prompts. Using the different topic terms, we utilise Wikipedia2Vec's directed network link-graph model to extract related entities and similar words using cosine similarity on Wikipedia2Vec embeddings. These Wikipedia2Vec embeddings will reflect articles, entities, and events available on Wikipedia up until the end of April 2022.

Figure 4.2 illustrates the relationship between parsing the pre-processed corpus and building the vocabulary using the above three feature extraction techniques.

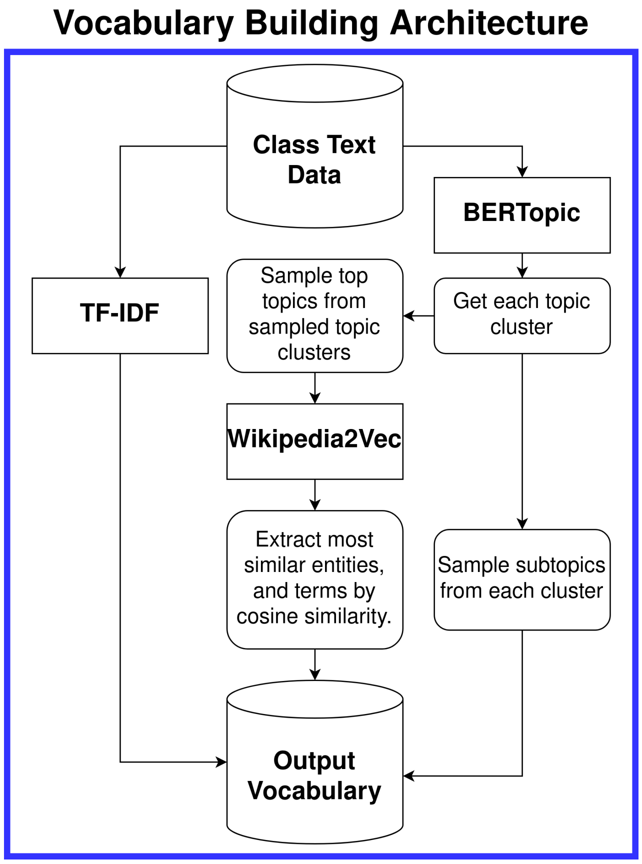

Vocabulary Building Architecture
Class Text Data
BERTopic
TF-IDF
Sample top topics from sampled topic clusters
Get each topic cluster
Wikipedia2Vec
Extract most similar entities, and terms by cosine similarity.
Sample subtopics from each cluster
Output Vocabulary

Figure 4.2: Architecture diagram for Prompt-GAN's vocabulary builder.

4.2.2.2 GPT-based Post Generator

We target models capable of local inference on our commodity setup, with 24GB VRAM and 32GB RAM. Targeting local models enables us to prove Prompt-GAN's capability without requiring expensive or high-memory models such as the 175 billion parameter GPT-3 model [14]. Furthermore, large online models have high power and cost requirements—with GPT-3's ~$0.12 per 1000 tokens for a fine-tuned model [74]. Hence, testing a local model for Prompt-GAN enables the architecture to remain fully open-source (unlike GPT-3) and resource/energy-constrained. We would expect more realistic text from expensive online models, as is the case with larger language models. However, Prompt-GAN seeks to prove the basis that even models capable of running on a single commodity computer can generate synthetic data which can replicate real hate patterns and thus help environmentally conscious and resource-aware researchers.

We target GPT-2 and GPT-Neo specifically as they represent two benchmarks in local language models. Prior fine-tuned models utilised GPT-2-large, a 768 million parameter model [107]. For Prompt-GAN, we test the full open-source GPT-2-XL and GPT-Neo-2.7B architecture which includes ~1.5 and ~2.7 billion parameters respectively [43, 80, 12].

To utilise the knowledge of recent tragic events such as the 2019 Christchurch shooting and the ongoing 2019-present COVID-19 pandemic, we also test the recent GPT-Neo models. GPT-Neo utilises architectural improvements from GPT-3 and GPT-2 to provide an open-source implementation of a GPT-3-like clone, trained on data up to late-2020 (exact cut-off date not specified) [38]. We test the GPT-Neo-2.7B model, containing 2.7 billion parameters and thus we would expect more realistic text given its larger parameter count.

We do not consider other language models such as T5, due to their lower runtime performance compared to a decoder-encoder architecture like GPT-2/Neo [67]. Likewise, we do not consider masked language model architectures such as BERT or BART for text generation—as GPT's auto-encoder architecture outperforms BERT-based models for open-ended text generation by the measure of accuracy and compute efficiency [14].

Table 4.1 outlines the considerable increases in parameter count (à la memory and computational requirements) for declining improvements in textual performance on the benchmark text generation datasets. We select GPT-2-XL and GPT-Neo-2.7B as parameter size considerably increases by over 10 billion parameters with diminishing returns—particularly given that GPT-3 would require ~116.67 times more memory for only ~10-20% higher accuracy scores. Furthermore, we ensure that Prompt-GAN can run on a commodity desktop environment to display that Prompt-GAN's architecture is both viable and capable of creating realistic synthetic online speech, and can remain resource, cost, and energy efficient.

Model	LAMBADA (PPL)	LAMBADA (ACC)	WikiText2 (PPL)	HellaSwag (ACC)	Winogrande (ACC)	PIQA (ACC)
BERT (110M)	-	-	69.32 [102]	38.30% [111]	51.90% [86]	66.80% [11]
GPT-2-Small (117M)	35.13 [80]	45.99% [80]	29.41 [80]	-	-	-
GPT-2-Med (345M)	15.6 [80]	55.48% [80]	22.76 [80]	-	-	-
GPT-2-XL (1558M)	8.63 [80]	63.24% [80]	18.34 [80]	40.03% [12]	59.40% [12]	70.78% [12]
GPT-Neo (125M)	30.266 [12]	37.36% [12]	32.285 [12]	28.67% [12]	50.43% [12]	63.06% [12]
GPT-Neo (1.3B)	7.498 [12]	57.23% [12]	13.1 [12]	38.66% [12]	55.01% [12]	71.11% [12]
GPT-Neo (2.7B)	5.626 [12]	62.22% [12]	11.39 [12]	42.73% [12]	56.50% [12]	72.14% [12]
GPT-3 Ada	9.95 [14]	51.60% [14]	-	43.40% [14]	52.90% [14]	70.50% [14]
GPT-3 (175B)	1.92 [14]	76.20% [14]	-	78.90% [14]	70.20% [14]	81.00% [14]
Jurassic-1 (178B)	-	-	-	79.30% [57]	68.90% [57]	81.40% [57]
PaLM (540B)	-	77.90% [17]	-	83.40% [17]	81.10% [17]	82.30% [17]
Human	-	-	-	**95.70%** [111]	**94.00%** [86]	**94.90%** [11]

Table 4.1: Performance on the shared datasets across the language models. Lower perplexity is better. Higher accuracy is better.

4.2.3 Discriminator

After the generator creates a prompt string to append to a static instruction prompt, we then use this prompt to generate 40 sequences of up to 100 tokens, or until the end-of-text token. We limit text generation to 100 tokens as this reflects approximately 400 letters, more than Twitter's 280-character limit to reflect longer Stormfront posts.

To avoid bias towards a specific class, we split all train and test data via equal-distribution stratified sampling to ensure an equal number of instances per class regardless of its distribution in the overall dataset.

We consider computational (RAM/CPU usage) performance, evaluation loss, and F1-score performance when selecting which variant of BERT to use as Prompt-GAN's discriminator—as outlined in our Experimental Design in Section 5.1.2.

After identifying the prompt string that generates the most realistic data by maximising discriminator loss, we then generate a synthetic dataset using five prompts that include different topic spaces. For instance, if the first generated prompt targets the topic "Ireland", we then remove any related terms to the "Ireland" topic cluster for the next iteration/prompt-generation. Thus, we ensure that new prompts use other topic clusters. We then employ Wikipedia2Vec on another unused topic cluster to find new entities and articles to drive the NLM towards new discussions for the synthetic posts.

4.3 Prototyping and Assumption Proving

Our first research question asks, "*Can neural language models produce **topic** and **platform-specific** hate speech with competitive F1-scores and toxicity metrics compared to a real hate speech corpus?*". Hence, the architecture of our Prompt-GAN model assumes that crafted prompts can increase the next token(s) probability towards a desired output. As a proof-of-concept, we consider biasing GPT-2/Neo towards a desired answer using a fixed static instruction prompt. If it is not possible to bias a neural language model towards a desired output, then we cannot construct synthetic hate and/or extremist-affiliated datasets to train our discriminator—thereby invalidating Prompt-GAN's architecture. We can test this assumption by biasing a fixed question with prepended tokens to aid a GPT-type model towards giving our desired answer.

Hence, in Section 4.3.1, we prototype and test an experiment to radicalise

GPT-2 by seeding its input prompt with relevant tokens to drive the model to complete the instruction prompt “Complete the sentence: My name is” with the objective to get the next token probability to be the answer of “Hitler”. We integrate and test BERTopic and Wikipedia2Vec to reverse-engineer the answer through a reverse query approach.

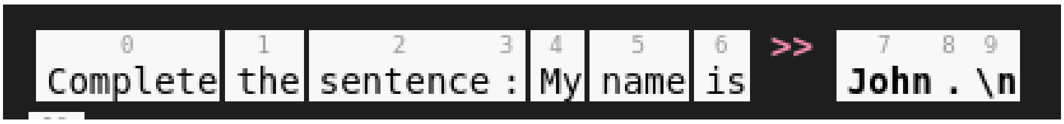
If we assume that seeded tokens can reliably create targeted output tokens, the next architectural challenge before constructing the full Prompt-GAN pipeline is identifying the unique topics and terminologies used in the hateful and non-hateful corpora. Our Prompt-GAN architecture assumes that the choice of words, terms and topics differ between classes whether they are explicitly mentioned, related or implied—such as for ideologies or latent beliefs. Subsection 4.3.2 explores the topic, term, and embedding-based clustering approaches to select the final methodologies used in our Prompt-GAN vocabulary builder architecture. We considered working on an embedding-level by prepending perturbed embedding vectors to the prompt layer. However, we found no notable cluster deviation between the hate and non-hate data when exploring numeric GPT and sentence-level embeddings. Conversely, TF-IDF and BERTopic demonstrated more apparent clusters between the hate/non-hate classes. Due to the lack of clusters between the hate and non-hate numeric embedding spaces, we reject prompt-tuning via perturbing numeric embeddings and instead consider a textual string-builder approach based on unique-to-the-class Wikipedia-derived entities, TF-IDF extracted terms, and related topics. Moreover, a prompt-tuning approach based on building textual prompt strings is also unique, given that the previous embedding-based approaches involved appending vector embedding via soft prompts [55], and perturbing numeric embeddings via p-tuning [60]. Subsection 4.3.2 includes the cluster visualisations and analysis of the contextual sentence-embeddings, while subsections 4.3.3 and 4.3.4 visualises and analyses the contrasts and topic-clusters between classes using the textual string-based approaches of TF-IDF and BERTopic.

4.3.1 Assumption Proving via Manual Exploration

In our first assumption-proving experiment, we consider whether prepending textual terms can increase and thus bias the next-token probability without changing the static instruction prompt.

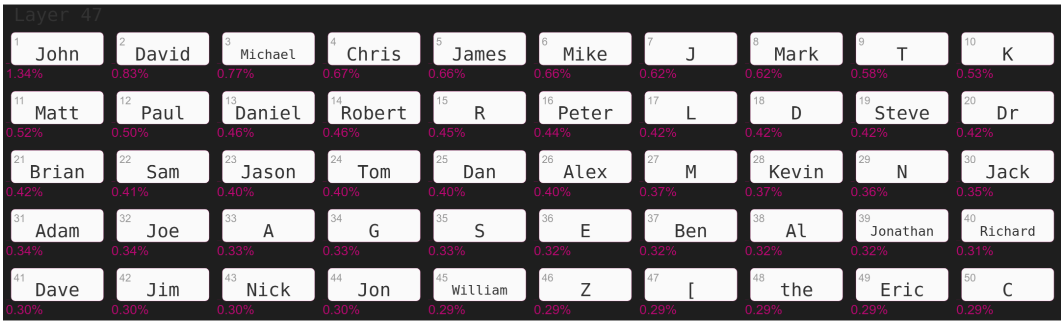
Our objective is to drive GPT-2's next-token probability to be that of "Adolf" or "Hitler"—due to his notoriety, prevalence in the data within the pre-trained corpus, and ideological relevance vis-à-vis far-right extremism. We test GPT-2-XL without token sampling and with the temperature parameter set to 0 to create replicable and deterministic next-token probabilities.

In the baseline *static prompt + sentence starter* example, the next-token probabilities reflect common Anglo-Saxon names—with Figure 4.4 demonstrating the Anglo-centric bias of GPT-2's pre-trained dataset. The next token probability for Adolf or Hitler is 3.284253e-4 and 8.151462e-05. Using these probabilities with top-k ranking (i.e., ordering the next token logits from the most probable to the least probable, across the full GPT token vocabulary), Adolf and Hitler would be the 546th and 1482nd ranked choice.



0	1	2	3	4	5	6		7	8	9
Complete	the	sentence	:	My	name	is	>>	John	.	\n

Figure 4.3: Baseline static prompt for GPT-2-XL (1.5B).



Layer 47

1 John 1.34%	2 David 0.83%	3 Michael 0.77%	4 Chris 0.67%	5 James 0.66%	6 Mike 0.66%	7 J 0.62%	8 Mark 0.62%	9 T 0.58%	10 K 0.53%
11 Matt 0.52%	12 Paul 0.50%	13 Daniel 0.46%	14 Robert 0.46%	15 R 0.45%	16 Peter 0.44%	17 L 0.42%	18 D 0.42%	19 Steve 0.42%	20 Dr 0.42%
21 Brian 0.42%	22 Sam 0.41%	23 Jason 0.40%	24 Tom 0.40%	25 Dan 0.40%	26 Alex 0.40%	27 M 0.37%	28 Kevin 0.37%	29 N 0.36%	30 Jack 0.35%
31 Adam 0.34%	32 Joe 0.34%	33 A 0.33%	34 G 0.33%	35 S 0.33%	36 E 0.32%	37 Ben 0.32%	38 Al 0.32%	39 Jonathan 0.32%	40 Richard 0.31%
41 Dave 0.30%	42 Jim 0.30%	43 Nick 0.30%	44 Jon 0.30%	45 William 0.29%	46 Z 0.29%	47 [0.29%	48 the 0.29%	49 Eric 0.29%	50 C 0.29%

Figure 4.4: Next token probabilities—demonstrating the pre-trained corpora's Anglo-bias with western names.

Underpinning this issue is the core difference between *semantic* and *topical* similarity. While Adolf is a name reduced in usage because of its historic namesake exploitation, an Anglo-bias would be to tie it to *Adolf Hitler* rather than benevolent Adolfs' or simply its usage as a Germanic name. Despite western Anglo terrorists, it is unlikely that an English-dominant speaker would associate "David" or "Dylan" to terrorists or far-right extremists of the same name. As our Prompt-GAN relies on automated approaches to find related terms, we cannot rely on word-by-word semantic similarity used by approaches like Word2Vec. To reverse search related terms and entities, we must consider a context-sensitive approach with "Adolf Hitler" being a singular *entity-based vector* rather than two independent words which do not convey the full meaning of the German leader.

In a similar manner to a search query, we utilise Wikipedia2Vec's link-graph model to identify related *contextualised entities* based on the extracted hyperlink connections between the related pages on Wikipedia to collect topical and ideologically relevant terms and entities.

In our reverse query of "Adolf Hitler", our updated Wikipedia2Vec model outputs the following related entities: *Nazi(sm), Nazi Party, Joseph Goebbels, Führer, Henrich Himmler, Hermann Göring, Mein Kampf, and Benito Mussolini.*

While BERTopic offers the topics of *Nazi*, *Jewish*, *Holocaust*, *Germany.*

While Word2Vec for *Adolf* offers German names such as *Franz*, *Josef*, *Johannes.* However, *Hitler* offers *Nazi*, *fascism*, *Reich* as three related words– likely due to the reduced use of Hitler as a name after the second world war.

Thus, a single token can increase the next token probability from 546th to 1st based on seeding the prompt with a topical context. Hence, we postulate that given the instruction to generate an online post, we can create *topical* and *hateful* text by substituting and prepending terms, topics, and names of entities to an instruction prompt. Having proved our assumption that prepended tokens can increase the next token probability towards a desired output, we proceed to consider experimenting with numeric-embeddings and extracting terms across a full hate speech corpora using TF-IDF, out-of-

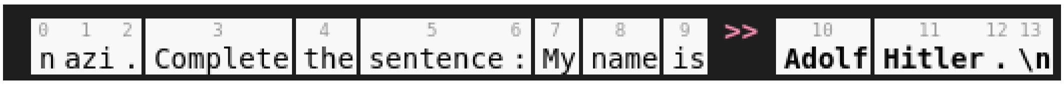
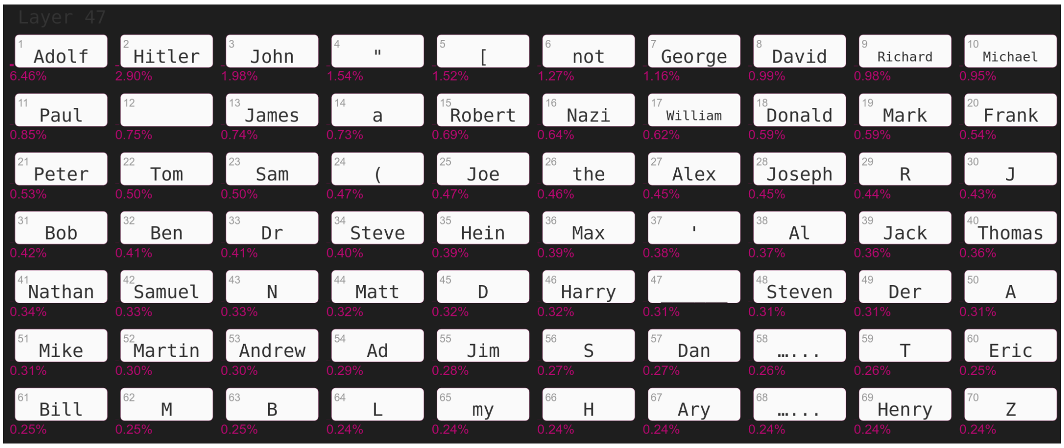

0 1 2 3 4 5 6 7 8 9 >> 10 11 12 13
n azi . Complete the sentence : My name is >> Adolf Hitler . \n

Figure 4.5: Extracted entity prepended to the static prompt for GPT-2 (1.5B).


Layer 47

1 Adolf 6.46%	2 Hitler 2.90%	3 John 1.98%	4 " 1.54%	5 [1.52%	6 not 1.27%	7 George 1.16%	8 David 0.99%	9 Richard 0.98%	10 Michael 0.95%
11 Paul 0.85%	12 0.75%	13 James 0.74%	14 a 0.73%	15 Robert 0.69%	16 Nazi 0.64%	17 William 0.62%	18 Donald 0.59%	19 Mark 0.59%	20 Frank 0.54%
21 Peter 0.53%	22 Tom 0.50%	23 Sam 0.50%	24 (0.47%	25 Joe 0.47%	26 the 0.46%	27 Alex 0.45%	28 Joseph 0.45%	29 R 0.44%	30 J 0.43%
31 Bob 0.42%	32 Ben 0.41%	33 Dr 0.41%	34 Steve 0.40%	35 Hein 0.39%	36 Max 0.39%	37 ' 0.38%	38 Al 0.37%	39 Jack 0.36%	40 Thomas 0.36%
41 Nathan 0.34%	42 Samuel 0.33%	43 N 0.33%	44 Matt 0.32%	45 D 0.32%	46 Harry 0.32%	47 ________ 0.31%	48 Steven 0.31%	49 Der 0.31%	50 A 0.31%
51 Mike 0.31%	52 Martin 0.30%	53 Andrew 0.30%	54 Ad 0.29%	55 Jim 0.28%	56 S 0.27%	57 Dan 0.27%	58 ...... 0.26%	59 T 0.26%	60 Eric 0.25%
61 Bill 0.25%	62 M 0.25%	63 B 0.25%	64 L 0.24%	65 my 0.24%	66 H 0.24%	67 Ary 0.24%	68 ...... 0.24%	69 Henry 0.24%	70 Z 0.24%

Figure 4.6: Next token probabilities—demonstrating the effectiveness of prepending textual prompts to drive Adolf to the top token probability.

corpus topics via BERTopic, and expanding our hateful topics to new domains via Wikipedia2Vec in the following subsection.

4.3.2 Embedding Clustering and Dataset Exploration

While skewing GPT-2 towards a singular token is largely non-trivial with as little as one token, paragraph-level contextual hateful tweets and Stormfront posts require more context within the prompt to create these synthetic posts. Hence, we scale up our prototyping with a new assumption—that the raw numeric embeddings from the GPT tokeniser and sentence-level embedding models (i.e., SRoBERTa and SBERT-MPNet) will generate embedding-spaces that delineate hate and non-hate classes.

For these visualisations, we parse the posts into the GPT-2 language model using the Huggingface pipeline library for feature extraction via pairwise cosine similarity. We pad the variable-length posts to the maximum embedding length across all posts for the dataset with zeros-based padding. Due to the high dimensionality of the output padded arrays, we only conduct dimensionality reduction using embeddings from the smaller GPT-2-medium (352M)

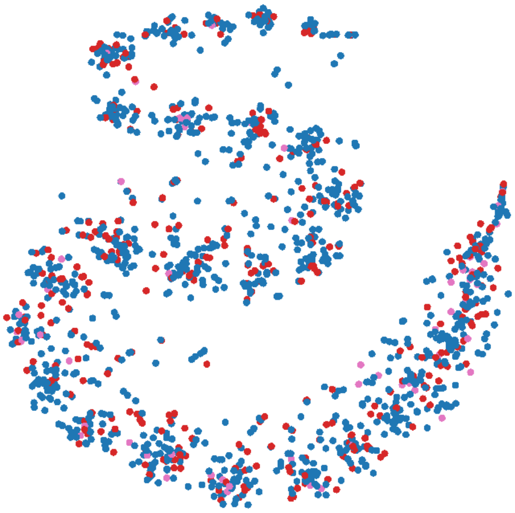


Figure 4.7: GPT-2 embeddings on the ElSherief et al. Implicit-Explicit Hate dataset. Red depicts *explicit hate speech*, pink depicts *implied hate*, and blue depicts *non-hate speech* tweets.

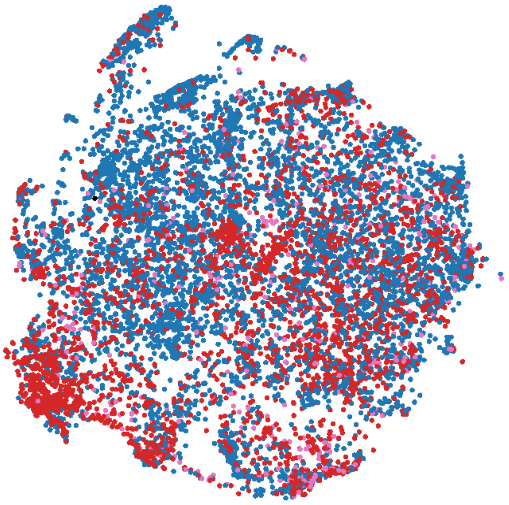


Figure 4.8: MPNet embeddings on the ElSherief et al. Explicit-Implicit Hate dataset. Red depicts *explicit hate speech*, pink depicts *implied hate*, and blue depicts *non-hate speech* tweets.

model. We do not consider truncating embeddings for longer posts, as it is not clear where in the longer-posts that the text contains actual hate speech—such as for long Stormfront posts. We conduct dimensionality reduction to visualise class clusters using t-distributed Stochastic Neighbour Embedding (t-SNE) and Uniform Manifold Approximation and Projection (UMAP) for both the GPT and SBERT embeddings. While the latter preserves the global structure of the dimensional manifold—UMAP's increased memory usage requires reducing the embedding size of the data. To avoid the aforementioned loss of information, we conduct UMAP only on the ElSherief et al. Twitter data—due to their 280-character limit compared to the variable-length Stormfront posts which would require more memory than available (32GB).

The lack of overlap between classes in both GPT and SBERT (semantic similarity) scenarios demonstrates that prompt-tuning with manipulated numeric embeddings is infeasible with current vector embedding models. Furthermore, a neural network approach for generating input numeric embeddings would further increase computational complexity due to the high dimensionality of GPT-2's input layer. Conversely, prepending tokens one-by-one enables

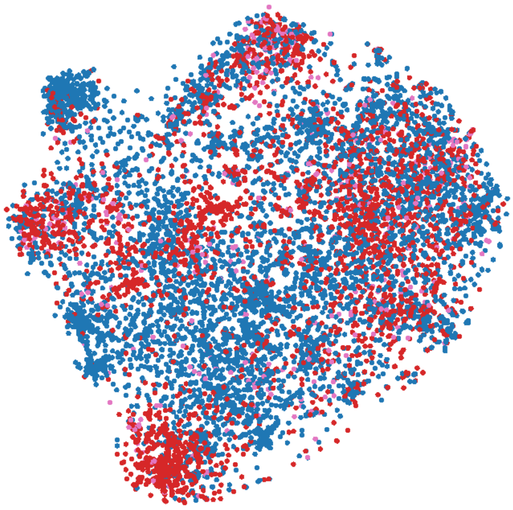


Figure 4.9: SRoBERTa embeddings on the ElSherief et al. Implicit-Explicit Hate dataset. Red depicts explicit hate speech, pink depicts implied hate, and blue depicts non-hate speech.

the retention of terms that increase evaluation loss for the discriminator (i.e., creating more realistic synthetic text), and the omission of terms that do not create more realistic text. Thus, we do not consider numeric embedding manipulation via alterable 'soft prompts' for our prompt-generator module.

4.3.3 TF-IDF in the Corpora

Non-hate Terms	Explicit Hate Terms	Implicit Hate Terms
white, people, racist, hate, black, like, right, race, just, don, **trump**, america, rt, **think**, anti, **antifa**, know, **alt**, **sure**, **supremacist**, **house**, **supremacists**, want, **media**	white, people, **kill**, jews, jew, racist, black, hate, **trash**, race, like, **faggots**, muslims, islam, america, stupid, just, don, whites, man, muslim, **blacks**, **cuck**, anti, want	white, people, black, race, racist, hate, america, like, jews, don, whites, anti, just, want, jew, **genocide**, man, **need**, **non**, right, muslims, **illegals**, **country**, islam

Table 4.2: Ranked TF-IDF terms on the ElSherief et al. dataset, displaying violent terms in the explicit hate data. Key non-overlapping words in bold.

Evidently, Table 4.2 demonstrates the linguistic styles between the classes, with non-hate data discussing regular politics and debating white supremacy in regard to the discussions on US politics with terms such as “Trump”, “antifa”, “supremacists”, and “media”. Conversely, explicit hate includes dehumanising slurs such as “faggots”, “cuck”, “trash” and a prominent use of “jews” as a demeaning category and stereotype, typically with violent connotations like “kill” as the third most common term. Conversely, implied hate speech tends to dehumanise “illegals” in their “country”, and tie discussions to imply “genocide”.

4.3.4 BERTopic Visualisation

RQ3 postulates: “*Can our model create synthetic online posts which target topics and other group affiliations outside of the training datasets?*”. To address RQ3, we must establish a baseline for the top topics discussed across the real corpora. Figures 4.10 and 4.11 depict the real hate and non-hate speech DG data, indicating the hate speech category’s use of slurs, anti-LGBT and anti-black sentiments; while the non-hate data discusses nationalism, geopolitics, weapons, events, perceived educational indoctrination, and topics pertaining to the *in-group* white supremacist *identity* (i.e., “whites”, and European people). Evidently, the non-hate data tends to target tangible assets and theory, while hate data tends to invoke aggressive personalised attacks towards those that are adversarial *out-groups* (i.e., “liberals”, “school” *institutions*, “homosexuals”, Jewish people, and people of colour via dehumanising slurs).

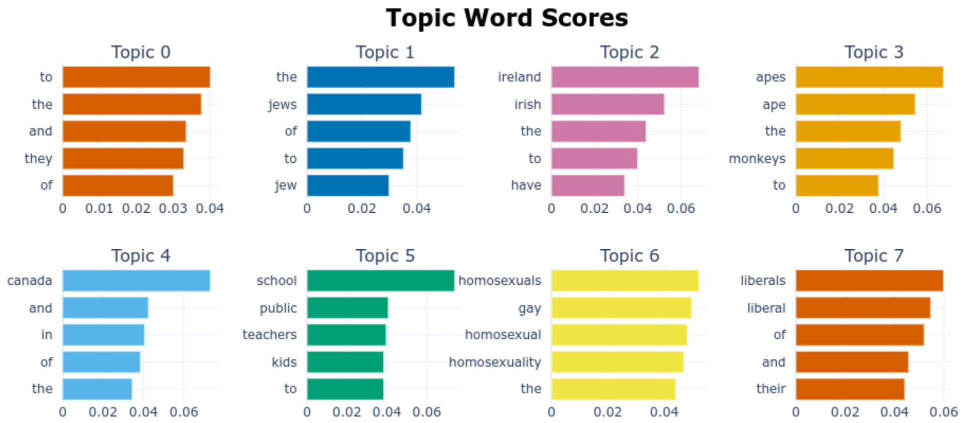


Figure 4.10: Extracted topic clusters from the de Gibert et al. Stormfront dataset's hate class posts.

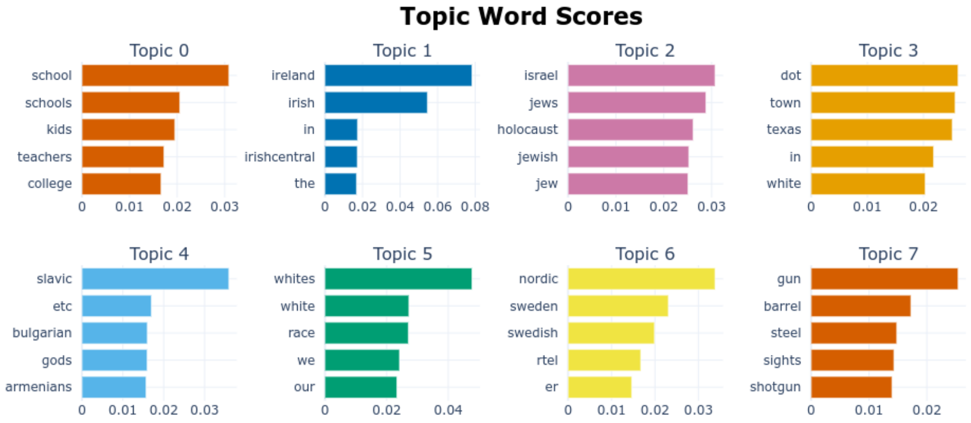


Figure 4.11: Extracted topic clusters from the de Gibert et al. Stormfront dataset's non-hate class posts.

Chapter 5

Evaluation

This chapter presents Prompt-GAN's experimental setup and results. Section 5.1 outlines our systematic approach to testing each component of the GAN—the textual prompt-generator, the language model, and the discriminator. We consider the full ERH Context Mining pipeline from Figure 2.5, including data selection and pre-processing pipelines, data filtering and feature extraction, and discrimination experiments on synthetic and real data. We conclude by presenting the outcome of our experiments in the Results Section 5.2. Further quantitative analysis and discussion of trends, implications, and usefulness of Prompt-GAN continue in the next Discussions Chapter 6.

5.1 Experimental Setup

In our experimental setup, we systematically test each component of our Prompt-GAN architecture as displayed in Figure 4.1. This section outlines the methodology conducted to generate the results displayed in graph and table form in the following Results section.

5.1.1 Experimental machine specifications

As outlined in our systematic literature review, there is a strong case for performance engineering with neural language models due to their prohibitive cost-per-token for online generative models such as GPT-3 or Jurassic-1. Furthermore, models with higher multi-billion parameter counts tend to outperform smaller models—although runtime, memory, and energy/green considerations are not discussed in prior work. As opposed to testing on distributed cloud systems (as required for large language models), we conduct the majority of our experiments on a local desktop with commodity hardware consisting of:

- **GPU:** 24GB RTX 3090—of which Video Random Access Memory (VRAM) is essential for testing larger language models for more realistic hate/non-hate generation.
- **CPU:** 3.7Ghz (base) to 4.8Ghz (boost) AMD Ryzen 5900X—only TF-IDF and Wikipedia2Vec methodologies run on CPU-based code rather than CUDA-based GPU code.
- **RAM:** 32GB DDR4-3600mhz.
- **OS/Drivers:** Lubuntu 22.04 (LTS) on Nvidia Driver 470.129.06 with CUDA 11.4.
- **Libraries used:** PyTorch v1.11.0, DeepSpeed v0.6.4, Ecco, scikit-learn v1.1.1, imbalanced-learn v0.9.1, Huggingface Library, Transformers v4.19.2, Tokenizers v0.12.1, NumPy v1.21.6, Pandas v1.4.2, Tensorboard v2.8.0, BERTopic v0.10.0, Wikipedia2Vec v1.0.5, Sentence-transformers v2.2.0.

Our max power draw is 460w during discriminator training. To train Prompt-GAN on two classes of data on one dataset would take ~44 minutes, thus using ~0.3373kWh. Once trained, text generation takes ~0.3-1.5 seconds per post, thus keeping power usage to generate a synthetic dataset to a minimum.

5.1.2 Data discrimination experiments

For all input data, we filter erroneous whitespace, non-ASCII characters, duplicate punctuation (replacing with just one instance, as GPT-2/Neo does not replicate such patterns), raw HTML code and website links as most links are from URL shorteners like bit.ly, which has no use of the data as we do not reverse search the links. As the de Gibert et al. dataset censors slurs and profanities, we reverse-engineer the slurs based on their structure (e.g., "bull * * * *" and "* * * * hole"). Although, this decensorship approach cannot guarantee a one-to-one matching for words of similar length or spelling. For speed and consistency, we implement the above pre-processing techniques via regular expressions when loading the dataset. We then tokenise the data using the respective BERT-derived transformer models, which we select based on the state-of-the-art classification performance as outlined on Huggingface's transformer documentation and related studies [43, 61, 25, 89, 42]. We pad the data using zeros-padding to the max token 512-token size for BERT-derived models [25] and truncate tokens beyond this max size—which is not required for all but one post with a character count of 2153 (~539 tokens), with only six posts with more than 1600 characters (~400 tokens). All baseline tests use the full dataset with class-proportionate stratified sampling for the train-test-split. Tables A.3, present the baseline performance on the de Gibert et al. (DG), Davidson et al. (DV) and ElSherief et al. (ES) datasets across the DistilRoBERTa [89], BERT [25], RoBERTa [61], DeBERTa [41], ELECTRA [18], and DistilBERT [89] models. We present additional baseline metrics for the binary forms of the ES dataset (merged explicit and implicit hate vs. non-hate) and DV dataset (hate and non-hate classes) in the Tables in Appendix A.5.

To optimise the fine-tuning process, we utilise incremental hyper-parameter tuning for top-k sampling, temperature, minimum output post word count, maximum output post word count, max token length, number of output prompts (with three to six prompts per class to generate diverse multi-topic datasets), and repetition penalties to avoid repetitious phrases and terms.

5.2 Results

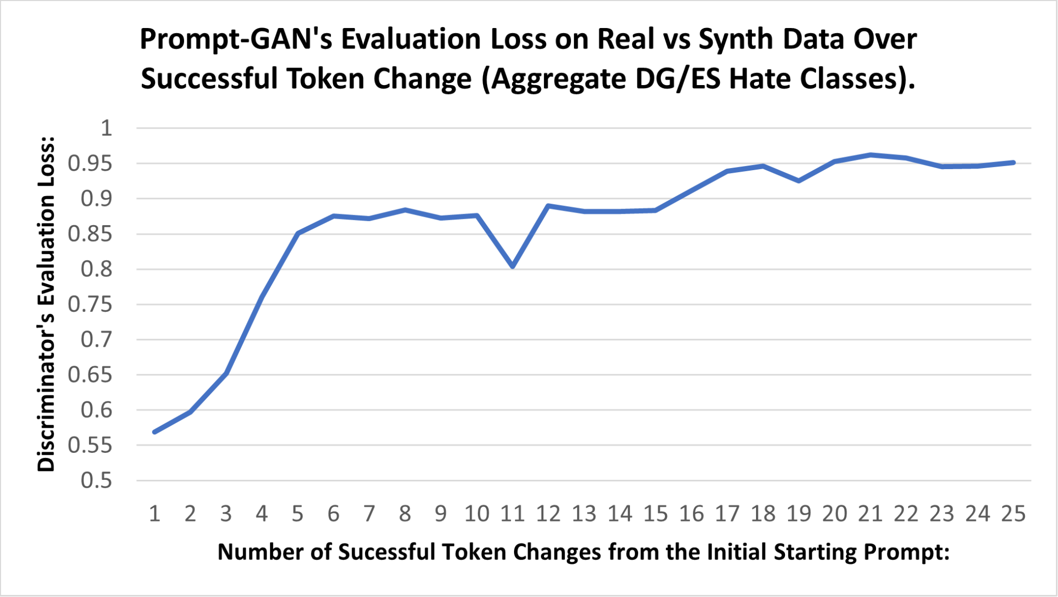


Figure 5.1: With each successful token addition or substitution to the static "Write a tweet:" prompt, the discriminator's uncertainty increases as the synthetic data becomes more *realistic* and *hateful*.

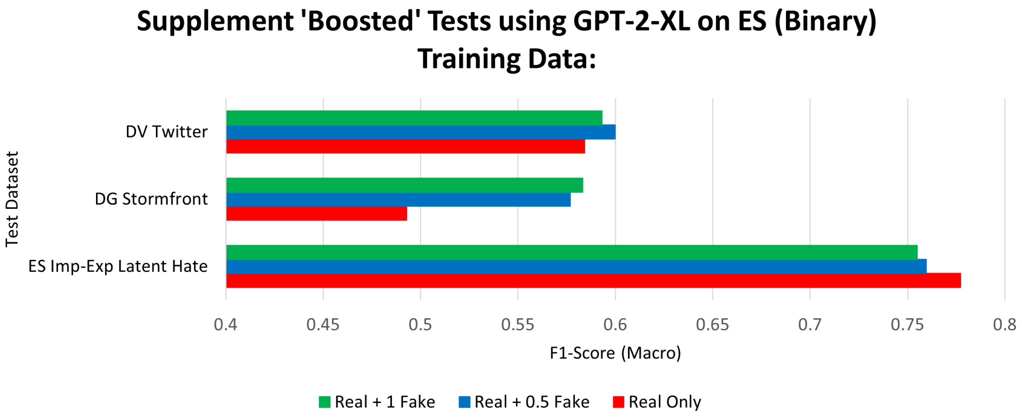


Figure 5.2: Supplement tests on the ElSherief et al. (ES) dataset (implicit & explicit hate speech merged as one binary class vs non-hate class), with F1-scores from predicting the ES, DG, and DV (hate & non-hate only) test data.

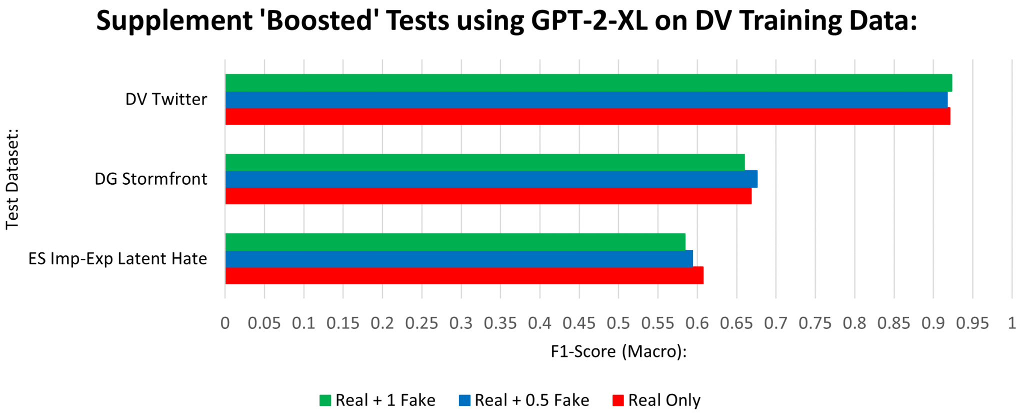


Figure 5.3: Supplement tests on the Davidson et al. (DV) dataset (hate & non-hate class), with F1-scores from predicting the ES, DG, and DV test data.

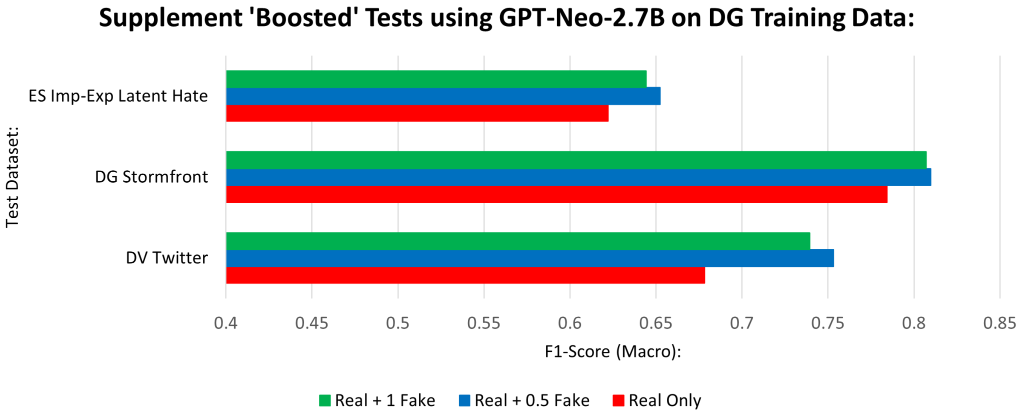


Figure 5.4: Supplement tests on the de Gibert et al. (DG) dataset, with F1-scores from predicting the ES, DG, and DV (hate & non-hate only) test data.

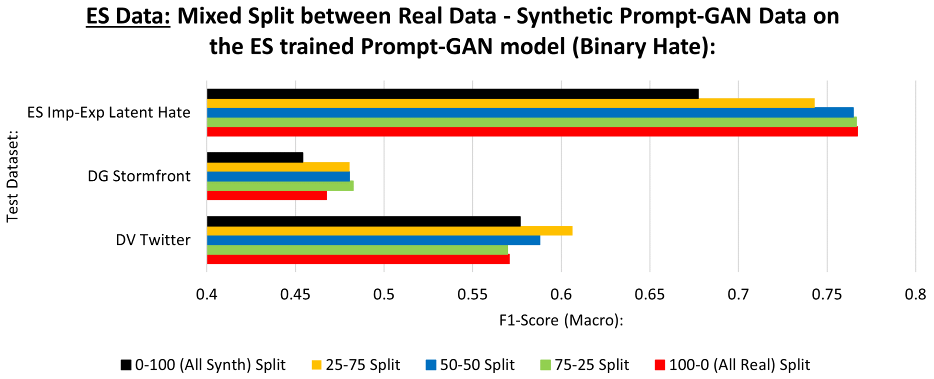


Figure 5.5: Mixed tests on the ElSherief et al. (ES) dataset, with F1-scores from predicting the ES, DG, and DV (hate & non-hate only) test data.

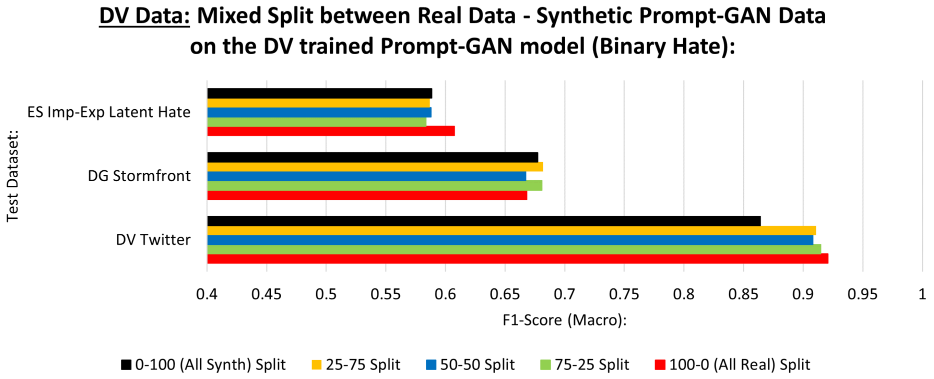


Figure 5.6: Mixed tests on the Davidson et al. (DV) *binary* dataset.

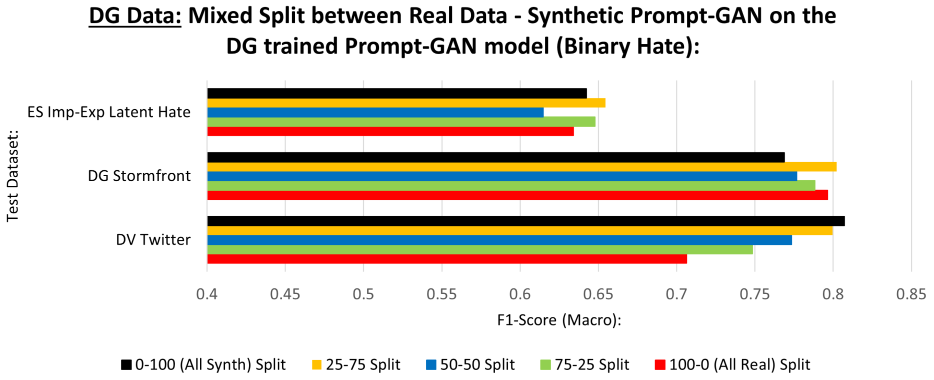


Figure 5.7: Mixed tests on the de Gibert et al. (DG) dataset.

Dataset	Data Class	All Synthetic Data's Mean Hate Probability	All Real Data's Mean Hate Probability
DG	Hate	0.7231	0.6153
DG	Non-hate	0.076	0.087
ES	Explicit Hate	0.5777	0.5083
ES	Implicit Hate	0.6423	0.6083
ES	Non-hate	0.0785	0.1407
DV	Hate	0.6544	0.5793
DV	Non-hate	0.0797	0.0835

Table 5.1: Testing Prompt-GAN and real data on the third-party BERTweet blind classifier [78], and calculating the mean hate probability from each post's classification. DV Offensive class excluded as it is not hate speech.

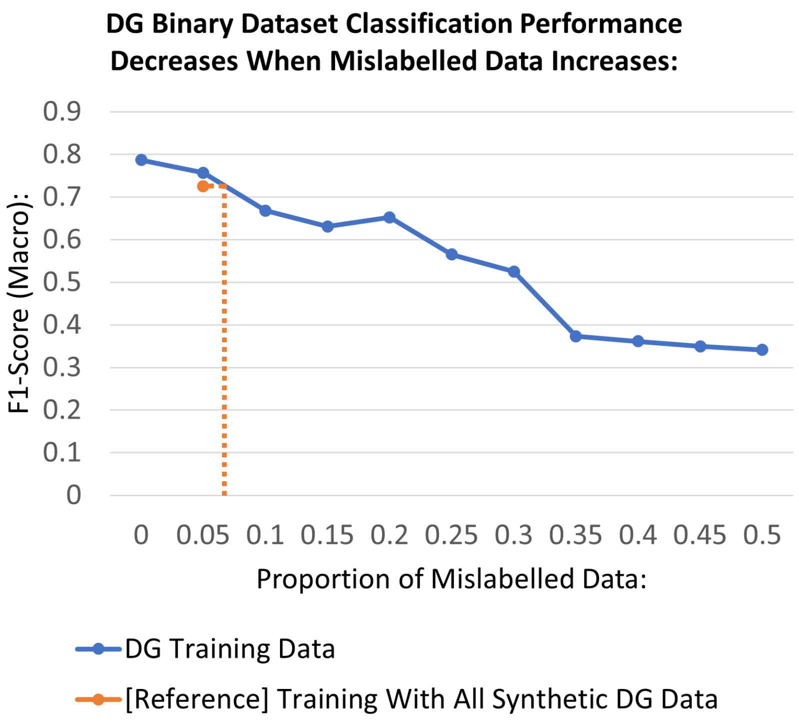


Figure 5.8: Prompt-GAN trained on the de Gibert et al. dataset (Hate or Non-hate) indicates that our synthetic data is ~93% correct to the class.

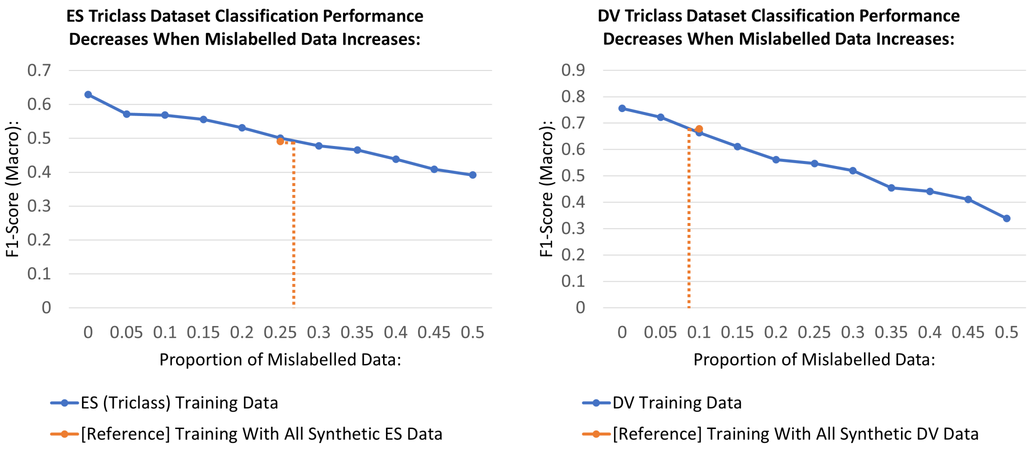


Figure 5.9: The discriminator's performance decreases with mislabelled data, useful for approximating Prompt-GAN's textual realism. The left and right graphs display the results from the ElSherief et al. dataset (Implicit hate, Explicit hate, Non-hate), and the Davidson et al. tri-class dataset (Hate, Offensive, Neither). Orange point displays the F1-score from the model trained only on synthetic DG posts and tested on the real DG test data.

Model	Dataset (Binary)	Recall (Macro)	Precision (Macro)	F1-score (Macro)
Prompt-GAN	**DV**	**94.0%**	**92.7%**	**93.3%**
Prompt-GAN (without K.D)	**DV**	**93.8%**	**94.0%**	**93.9%**
Prompt-GAN (deberta-base)	**DV**	**95.2%**	**93.9%**	**94.6%**
Our baseline (all-real DV data)	DV	92.9%	91.0%	91.9%
Wullach et al. MegaSpeech [107]	DV	81.4%	92.3%	86.5%
Prompt-GAN	**DG**	**81.0%**	**80.9%**	**81.0%**
Prompt-GAN (without K.D)	**DG**	**81.8%**	**78.9%**	**80.2%**
Our baseline (all-real DG data)	DG	75.8%	75.3%	75.6%
Wullach et al. MegaSpeech [107]	DG	58.2%	60.0%	59.1%
Baseline DG model [23]	DG	73%	-	-

Table 5.2: Comparison of synthetic data supplement experiments on DG and DV data. Only hate and non-hate class included for the DV results, as Wullach et al. [107] drop the Offensive class tweets. We select the highest F1-scoring real-synth mix from our experiments, and the highest F1-score result from the related studies. K.D = Knowledge Distillation (distilled discriminator).

Model	Dataset (3-class)	Recall (Micro)	Precision (Micro)	F1-score (Micro)
Prompt-GAN	**DV**	**91.6%**	**91.6%**	**91.6%**
Prompt-GAN (without K.D)	**DV**	**92.0%**	**92.0%**	**92.0%**
Our baseline (all-real DV data)	DV	91.0%	91.0%	91.0%
HateGAN (Cao and Lee [16])	DV	-	-	89.6%
Baseline DV model [21]	DV	-	-	90.0%

Table 5.3: Comparison of *multi-class* synthetic data supplement experiments on the DV data. Micro F1-score used as macro F1 scores are not provided by the other studies. K.D = Knowledge Distillation (distilled discriminator).

5.2.1 Topic modelling on the synthetic text

Using BERTopic, we identify and visualise the topic clusters from the most to least important for the top eight topics with their topic (y-axis) and topic significance score based on cosine similarity (x-axis). Each visualisation represents a specific class for one of the three datasets used to train Prompt-GAN. In our discussions, we cross-examine the in and out-of-domain topics in the synthetic data compared the baseline real data.

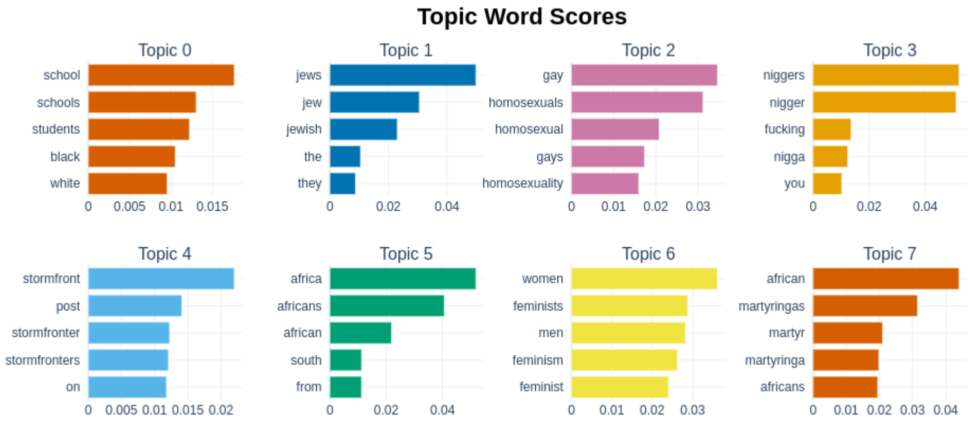


Figure 5.10: Extracted topic clusters from the synthetic hate class data via Prompt-GAN trained on the de Gibert et al. Stormfront dataset.

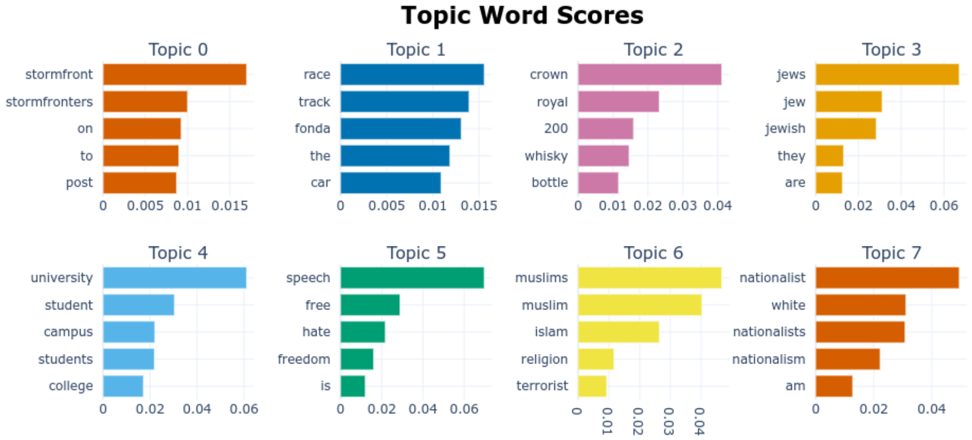


Figure 5.11: Extracted topic clusters from the synthetic non-hate class data via Prompt-GAN trained on the de Gibert et al. Stormfront dataset.

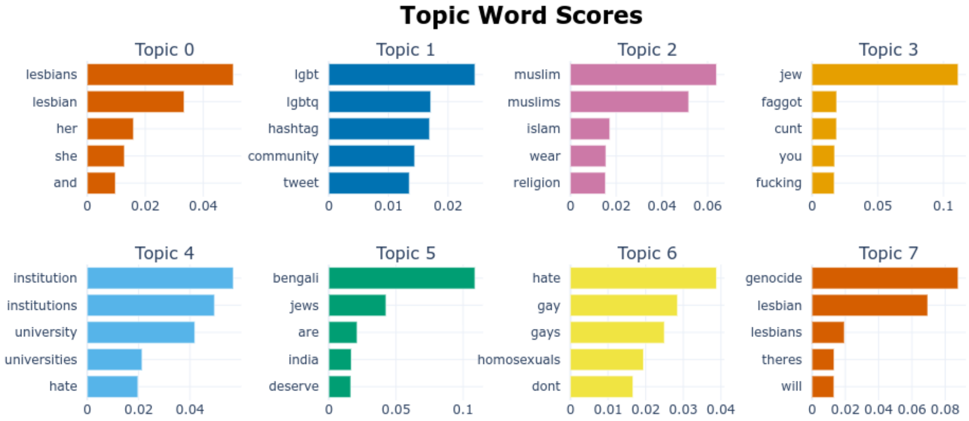


Figure 5.12: Extracted topic clusters from the synthetic implicit hate class data via Prompt-GAN trained on the ElSherief et al. Twitter dataset.

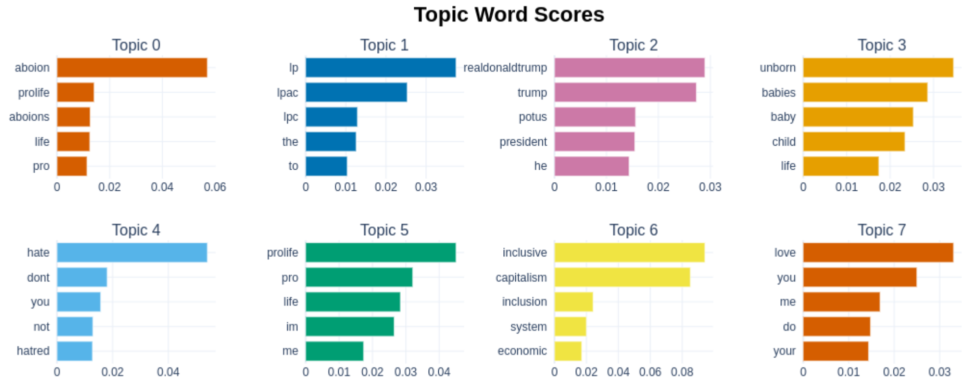


Figure 5.13: Extracted topic clusters from the synthetic neutral non-hate class data via Prompt-GAN trained on the ElSherief et al. Twitter dataset.

Synthetic Example Stormfront (DG) or Twitter (ES/DV) *hate* posts	**Label**	**Source**
*"you did a statement every night fag*ot you got fag*ot [sic]"*	Hate	HateGAN [16]
*"is it a crime we know the true americans are isis just fuc*ing id*ots [sic]"*	Hate	HateGAN [16]
"The number of refugees and immigration controls are clear, now is it time to go to extremes in a desperate attempt to stop them."	Hate	GPT-2 Fine-tuned on DG data [107]
"the left celebrates the death of freedom as it is the road to the extermination of whites and the rebirth of the baboon"	Hate	GPT-2 Fine-tuned on DG data [107]
"The Jews were behind the Bolshevik Revolution and the Holocaust. They control Hollywood, the media, academia, and virtually all government positions in the west"	Hate	Prompt-GAN (Stormfront)
"Cultural Marxism has no place in the White Nationalist movement. It is the destruction of the European family. They are just anti-white, plain and simple (or antiwhite.)"	Hate	Prompt-GAN (Stormfront)
"I heard that blacks are lazy and dumb as rocks. So I decided to put a few black niggers into a bag and let them roam around on my block because I guess nigga is a dirty word."	Hate	Prompt-GAN (Stormfront)
"Hey @piersmorgan you're a queer, a man, and your show is about as relevant as a wet napkin in a bar. #pussyhat"	Imp. Hate	Prompt-GAN (Twitter, ES)
"Your country is full of rapists, bastards and drug dealers. People like you should be killed by your own people and those that would rather live with you in peace. I hope it is your fucking time. #MakeAmericaRapeAgain"	Exp. Hate	Prompt-GAN (Twitter, ES)
"Did you know that if You don't have any money, you are still able to have sex? @realDonaldTrump, take that!"	Offensive	Prompt-GAN (Twitter, DV)

Table 5.4: Synthetic hate class data from HateGAN (trained on Twitter), GPT-2 fine-tuned on DG dataset (MegaSpeech), and our Prompt-GAN model.

Synthetic Example Stormfront (DG) or Twitter (ES/DV) ***non-hate*** **posts**	**Label**	**Source**
"thanks for your response this is what our school system is becoming"	Non-hate	GPT-2 Fine-tuned on DG data [107]
"A member of the Liberty Bell Choir sings the National Anthem before the game between the New York Yankees"	Non-hate	GPT-2 Fine-tuned on DV data [107]
"We teach about free speech at school, but how about we be a tad more liberal and take a stand against hate speech and white supremacy?"	Non-hate	Prompt-GAN (Stormfront)
"@HuffingtonPost: You don't need a Phd. to see that @BarackObama doesn't deserve a @POTUS title."	Non-hate	Prompt-GAN (Twitter, ES)
"The people of Turkey stand united with our Kurds, with the people in the West who stand with all people who are being massacred for their beliefs. Terrorizing and killing innocent people by the government only serves the terrorists."	Non-hate	Prompt-GAN (Twitter, ES)
"Feminism and Communism is just one small example of what the left is all about. The communists hate that I am not a capitalist, and the feminists only hate male chauvinists who don't pay for child care. (Funny, if their beliefs and behavior wasn't so damn hypocritical)."	Non-hate	Prompt-GAN (Twitter, ES)
"Dear @realDonaldTrump, your proposed Muslim ban violates @POTUS's commitments to defend #US principles: Freedom of Religion, Freedom from Religion."	Non-hate	Prompt-GAN (Twitter, DV)

Table 5.5: Synthetic non-hate class data from HateGAN (trained on Twitter), GPT-2 fine-tuned on DG and DV dataset (MegaSpeech), and our Prompt-GAN model.

Chapter 6

Discussion

Our results for Prompt-GAN consist of three fundamental analytical elements:

1. The investigation of synthetic data to help supplement (*boost*) or substitute (*mixed*) real posts in a dataset for hate speech classification, in comparison to the baseline models trained on the original real data.

2. The classification values of our synthetic data in comparison to the real data when applied to a third-party hate classifier which was not trained on the real or synthetic data used in our study (i.e., a *blind* test).

3. The linguistic analysis of topics and group affiliations, alongside readability and technicality metrics between the real and synthetic data. Our socio-linguistic analysis addresses RQ3's requirement to identify if Prompt-GAN's topics and group affiliations include those that are not part of the original dataset per our *domain expansion* approach.

We further back our findings through our proposed suite of textual realism metrics to form a *digital Turing test.* To address RQ1, we go beyond just cross-examining classification performance on the original dataset by identifying linguistic and topical trends, as well as its utility for transfer learning based on the out-of-domain topics imbued within our *domain expansion* strategy. For this *digital Turing test*, we stratify our experiments to cover the synthetic data's classification performance when supplementing or replacing text in the training data in Subsection 6.1.2, alongside hate scores from the independent third-party model trained on a non-overlapping Twitter corpus [78]. To address RQ3's out-of-corpus/new discussion capability, we analyse linguistic properties via topic modelling and technicality scores of the Automated Readability Index and Flesch-reading ease score [90, 51, 24].

6.1 A Digital Turing test—Results and its Implications for the Research Questions

Research question one asks: *"Can neural language models produce topic and platform-specific hate speech with competitive F1-scores and toxicity metrics compared to a real hate speech corpus?"*. Hence, this research question seeks to prove if Prompt-GAN can create types of hate speech that can substitute or aid (*boost*) real datasets. We do not expect a model trained on all or mixed synthetic data to outperform the real data on the same test corpus for the simple reason that it is not possible to create synthetic data that is more realistic than the real data. We hypothesise that supplementing a real dataset with synthetic data should not taint F1-score performance—which we quantify as a greater than 5% reduction in test F1-score. We expect any synthetic data generation model to include false positives as none of the GPT-derived models achieves 100% accuracy in any of the language model benchmarks aforementioned in Table 4.1) nor outperforms humans at this point in time. We predict that our synthetic data will improve classification performance on the external datasets (per RQ2) due to our approach of exploring terms, entities, events, and individuals using our *domain expansion* approach. Our domain expansion approach includes querying related concepts from the extracted topics from the real data, and reverse searching related terms from BERTopic and entities from Wikipedia2Vec.

We target the F1-score as the most important classification metric—as the datasets are class-imbalanced and thus a high accuracy could simply reflect a bias towards a majority class, which would lead to low precision. F1-score represents the harmonic mean between precision and recall to balance correct classifications for a class overall (recall) and correct classifications across each group (precision). For instance, the de Gibert et al. dataset includes 1010 hate-class full posts, and 3983 non-hate full posts, where we define a "full post" as all labelled sentences of a post where if one sentence in the post has a hate label, then the entire post is considered *hateful*. Hence, a classifier that resorts to flagging all data as hate will attain a high 0.7977 recall for hate class classification, but a low 0.2023 precision for the overall dataset. This accuracy-precision imbalance occurs as the discriminator incorrectly classified all of the non-hate class data. F1-score counteracts this by accounting for recall and precision, with *macro* F1-score weighing each class equally—ideal for our imbalanced training datasets. We only consider the real data for our test datasets, as separated from the real training data via our stratified train-test split on the ES, DV, and DG datasets.

We investigate the impact that mislabelled data can have on classification performance, with Figures 5.8, 5.2 and 5.2 showing how classification performance decreases as incorrectly labelled data increases, whereby a training dataset with over 30% incorrect data results in F1-scores as low as ~0.35. When we train Prompt-GAN and use its synthetic data to train the classifier and test on the real DG test split, we attain an F1-score of 0.725. A 0.725 F1-score is approximately equal to, or the experiments where we trained the distilroberta classifier using only the real training data but with the labels flipped for 5% and 10% of instances in the dataset to simulate mislabelled data expected by any synthetic GAN model. With 5% mislabelled DG data, we attain a 0.757 F1-score, while 10% mislabelled real data attains a 0.668 F1-score, as visualised in Figure 5.8. Hence, we can reasonably conclude that Prompt-GAN generates a correct to the class hate or non-hate Stormfront post ~93% of the time (macro F1-score). Using this approximation approach, we also estimate that Prompt-GAN creates correct to the class data ~91% of the time for the tri-class *Offensive*, *Hate speech*, and *Non-hate* DV dataset, and ~74% for the nuanced tri-class *Implied hate*, *Explicit hate*, and *Non-hate* ES dataset.

6.1.1 RQ1: Baseline results from the real-data only discriminator

It is essential that the discriminator can detect and converge towards identifying the desired class—in our case hate categories to train Prompt-GAN. Utilising neural language transformers presents the state-of-the-art for hate speech classification. As Prompt-GAN must simultaneously and efficiently load and run both a text-generating NLM and a discriminating model within memory, we must balance classification performance with memory utilisation.

To establish a baseline to answer RQ1's *competitive F1-scores* requirement, we select distilroberta-base with a batch size of 40 as the discriminator for our GAN for its high classification performance and low VRAM usage—with the full results in the Appendix A.1. The additional ~1.5% F1-score of the roberta-base model on the de Gibert et al. data does not justify the additional 4.5GB of VRAM and the near double training time—resources which could otherwise be allocated to a larger text-generation model. Likewise, larger models with lower batch sizes underperform due to overfitting the smaller batches, while too large of a batch size results in overfitting and high memory usage. We also generate the prompt token vocabulary before loading in the generator and discriminator models to reduce peak memory usage.

Across all models using only the real data, we experience reduced classification performance on the most challenging multi-class task—implicit hate, explicit hate, and non-hate detection. This is expected as implied hate includes latent geopolitical and historic themes that imply hatred, such as neo-colonialism, belittling based on protected characteristics, and straw-manned arguments founded on discrimination [28].

6.1.2 RQ1/2: Results from the Prompt-GAN synth text

In this subsection, we dissect RQ1's requirement for competitive F1-scores by analysing the supplemented dataset experiments using the full real training dataset with 50% or 100% more posts from our synthetic corpus. We also conduct mixed data experiments, which substitutes real data with the synthetic data in increments of 25% of the dataset; and all synthetic training experiments. In all tests, we present the type of GPT model (i.e., Prompt-GAN using GPT-2-XL or GPT-Neo-2.7B) based on whichever has the higher mean F1-scores from each dataset.

For supplementing the training data with an equal proportion of binary hate and non-hate instances, Figures 5.2, 5.3, and 5.4 demonstrate that additional synthetic data does not notably taint/reduce the performance by more than 1.7% in the case of the ES dataset—below our 5% threshold. This result corroborates with our early approximation that the prompts generated from training Prompt-GAN on the ES dataset reflect their hate or non-hate class 74% of the time. Thus, the reduction from 0.777 (all-real training data) to 0.7599 (all-real + 100% of the synthetic ES binary hate data) reflects this slight shift given the mislabelled Prompt-GAN data. In the DG dataset, we experience an unexpected improvement in the DG test dataset performance, with a 1.9% higher F1-score with the addition of 50% of the synthetic DG data. This improvement is unexpected as we assumed that the synthetic data cannot outperform the real data from the same overall dataset. This improvement is likely due to the unique out-of-corpus topics from our domain expansion method, which expands the overall knowledge and talking points of the synthetic data. For instance, Figures 4.10 and 5.10 depict the real and fake DG dataset's topics—with our Prompt-GAN model including new discussion topics including chauvinist/anti-feminist discussions, martyrdom, and education. These additional topic clusters add additional examples of hate besides racism, namely sexism and perceived cultural Marxist indoctrination alongside race-mixing in the context of education. We present the raw output from the synthetic hate prompts in Table 5.4, with an example of the cultural Marxism conspiracy theory with undertones of the Great Replacement theory per its references to the “destruction of the European family” in context to an “anti-white” force.

For the mixed DG real-synthetic experiments, we observe an impressive case where the synthetic Stormfront posts met or exceeded the baseline discriminator. Figure 5.7 displays how our mixed data *outperforms* the baseline real data, including for the DG test data. Moreover, we observe a considerable F1-score increase in our all-synthetic data experiment on the DV test dataset, attaining a 0.8072 F1-score compared to the baseline all-real model’s 0.7063 F1-score—a significant 10.1% classification F1-score increase. Given that the F1-scores for the DG data are approximately equal to or greater than the baseline real data model, then we can conclude that Prompt-GAN can create realistic Stormfront posts that match the real data’s patterns.

Likewise, the performance of the mixed and all synthetic data experiments on the DV and ES datasets indicated approximately equal scores within a 1-5% F1-score variance, as visualised in Figure 5.6 for the Prompt-GAN DV data, and Figure 5.5 for the Prompt-GAN ES data. Prompt-GAN’s largest challenge emanates from its ~9% F1-score reduction when using the all synthetic ES training data, likely due to the low F1-score of the baseline all-real discriminator on the ES data. Moreover, the lower performance of both our all-real baseline ES discriminator and the Prompt-GAN mixed and all synthetic models demonstrate the ongoing challenge of identifying what draws the line between subtle implicit hate and non-hate speech.

Our results exceed the existing state-of-the-art in all F1-score metrics in comparison to the fine-tuning approach by Wullach et al. [107], and the LSTM model by Cao and Lee [16]. Our results improve on the existing state-of-the-art by up to 21.9%, as shown in Table 5.2 for the binary data, and Table 5.3 for the multi-class DV data. No prior text generation studies considered using the ES dataset.

Furthermore, we demonstrate that Prompt-GAN's data is approximately as hateful as the real data through a third-party *blind* classifier from Pérez et al. [78]. For this third-party BERTweet hate scorer, Table 5.1 demonstrates a clear ~0.5-0.65 mean class probability delta between the hate and non-hate classes—proving that Prompt-GAN's synthetic hate speech data is demonstrably hateful. Likewise, both the real and synthetic Stormfront non-hate speech posts are distinguishable with a 0.076 mean hate probability, similar to the 0.087 mean hate probability for the real non-hate speech Stormfront posts.

Overall, we can be confident in Prompt-GAN's performance knowing that the synthetic data's risk of false positives is not significant to reduce classification performance dramatically vis-à-vis the 5% F1 score threshold. Furthermore, we observe a negligible less than 5% F1-score deviance when we substituted posts in our mixed real-synthetic experiments and tested on the same dataset. In our supplement boosted dataset, our *increased* F1 score on the DG test dataset demonstrated how additional data can increase discriminator performance—likely due to GPT-2's knowledge of online discussion culture and political knowledge from its pre-trained online corpus.

6.1.3 RQ2: Can our model generalise to other datasets via transfer learning?

With RQ1 demonstrating comparable and competitive performance on the same dataset, another key strength of Prompt-GAN is its ability to create *generalisable* multi-class posts vis-à-vis RQ2. The real-data models suffer a significant drop in F1-score performance when classifying another dataset and other platforms (Stormfront or Twitter). In all tests with Prompt-GAN data, classification F1-scores were approximately equal to, or outperformed, the real-data-only model. Adding the ES-data trained Prompt-GAN data to the real ES dataset led to a considerable improvement in F1-score classification on the blind (unseen by Prompt-GAN and not part of the ES dataset) Stormfront DG dataset—leading to an 8.2% higher test classification F1-score.

A considerable improvement also occurs when adding Prompt-GAN synthetic posts to the DG binary dataset, leading to a 6.1% higher F1-score on the unseen DV dataset as visualised in Figure 5.4. The F1-scores differences between the *baseline* all-real and *boosted* real-synth datasets result in improvements higher than those experienced when testing different discriminator models. For instance, the state-of-the-art larger roberta-base model outperforms the older bert-base and smaller distilroberta-base model F1-scores by only 2-3% per our baseline tests. Thus, our results demonstrate how advances in deep learning need to focus more on improving the datasets themselves—including via data synthesis via our Prompt-GAN model, rather than solely focusing on larger and more computationally expensive discriminator models. Furthermore, we can confidently state *that our model is generalisable* and thus RQ2 holds based on the aforementioned mixed real-synth experiments, with Figure 5.7 demonstrating a 10.1% higher test F1-score on the test DV dataset, using the discriminator trained on the real-synth DG data; and 8.3% higher test F1-score on the test DG Stormfront dataset, but trained on the boosted ES real-synth Twitter dataset—visualised in Figure 5.2.

Therefore, we can confidently conclude that we have set a new benchmark for synthetic text generation across datasets in regard to RQ1 (i.e., *competitive* F1-score performance on the same test dataset), and RQ2 (i.e., the synthetic data's knowledge, topics, and hateful content *generalises* to the other datasets through higher classification performance compared to a model trained on only real data).

6.1.4 RQ3: Analysis of Prompt-GAN's Topical and Linguistic Diversity

To identify if Prompt-GAN can adapt to future hate speech datasets, we ask the question: "*Can our model create synthetic online posts which target topics and other group affiliations outside of the training datasets?*" (RQ3).

We observe multiple out-of-corpus topics, entities, and events within our prompt vocabulary and final prepended tokens. Our domain expansion approach expands our vocabulary through sampling topics from the corpus, extracting related words and topics from BERTopic, and querying Wikipedia's entities via Wikipedia2Vec's link-graph model and related words across the word-embedding space (akin to Word2Vec). Prompt-GAN then automatically prepends these words with contextual stop-words and TF-IDF terms to create prepended keyword strings.

For instance, final non-hate speech prompts included references to radical action groups such as the "Rose City Antifa", which encompasses the Portland Oregon action group from the leftist Antifa (an abbreviated moniker for Anti-fascist Action, based off the 1932-33 German anti-fascist 'Antifaschistische Aktion' group) [69]. References to far-right groups appeared in both the synthetic non-hate and synthetic hate speech posts, as seen in Prompt-GAN's selected vocabulary terms of 'atomwaffen', referencing the far-right neo-Nazi Atomwaffen Division [93]; and 'Groypers', referencing a loose online network of US alt-right white supremacists [2].

Prompt-GAN's prepended tokens typically mimic the underlining political and social themes of the original dataset—with Figure 5.10 and Figure 4.10 demonstrating the topics from the DG Stormfront synthetic hate and real hate. Both synthetic and real data share key anti-black racism, LGBT+ discrimination and dehumanisation of minorities. However, the synthetic data includes a unique anti-feminist/chauvinistic cluster—which is relevant with the rise of anti-woman "incels". Incels reflect a recent rise in anti-woman attacks propelled through online forums, with the Anti-Defamation League framing incels as a "subset of the online misogynist "manosphere" that includes Pick Up Artists and Men's Rights Activists, incels are known for their deep-seated pessimism and profound sense of grievance against women" and are "the most violent sector of the manosphere" [3]—also backed up by the Isla Vista shooter's incel motive [36]. Synthetic incel-aligned Stormfront posts tend to espouse a perceived moral degradation of women and highlight white male supremacy. Likewise, the synthetic implied hate speech category from the Prompt-GAN ES-data references rape and genocide towards female groups and sexual minorities—merging anti-female and anti-LGBT sentiments as visualised in the synthetic topic clusters in Figure 5.10. Our quoted example from Prompt-GAN via GPT-2-XL in Table 5.4 displays an educational indoctrination theme by claiming that a Jewish elite "control academia, the media, and virtually all government positions in the west"—reflecting the anti-semitic cultural Marxism conspiracy theory [95].

A final noteworthy group affiliation outside of the real datasets were Indian political prompt tokens highlighting regional ethnic groups. Figure 5.12 depicts the synthetic implicit hate speech Prompt-GAN data, with a topic cluster targeting Bengali people. Furthermore, the vocabulary from Prompt-GAN reflects this pivot towards international non-Western politics with references to groups such as 'Shiv Sena', an ultranationalist Indian party; 'BJP'—the Bharatiya Janata Party, one of the two major Indian parties; and 'Modi', the current (as of June 2022) Prime Minister of India from the BJP party. All of these terms exist in the synthetic topic clusters, as well as within the prepended tokens for the tuned prompt. Prompt-GAN also reflects Twitter-specific features, such as pseudo-links, Twitter handles, and hashtag trends.

It is also possible to manually seed the vocabulary generator to create specific topics and affiliations. In Table 6.1, we present an example of synthetic political tweets surrounding the 2022 Russian invasion of Ukraine as a starting topic, and expand Prompt-GAN's prompt vocabulary via the updated Wikipedia2Vec model to extract and query related entities, events, and terms to prepend to the static prompt "*Write a pro-[Russia, Ukraine] tweet:*". Since GPT-Neo-2.7B has no knowledge beyond its 2020 corpus, it still includes inaccuracies in relation to politicians and knowledge. The references to Donetsk and Luhansk refer to the separatist conflict zones at war with the Ukrainian armed forces since 2014 [52].

Synthetic Text	Class label
"The warplanes of the Ukrainian army will not lose in the face of invaders. We will not let you seize this territory... Fight bravely for the freedom of Ukraine... For Ukraine!"	Pro-Ukraine
"We are working to identify and isolate the pro-Russian forces, but our first priority is to protect the civilians of Donetsk and Luhansk. We will not allow the blood of our brave soldiers to be used to blackmail eastern Ukraine."	Pro-Ukraine
"@NATO: The US should re-think its aggressive policies toward Ukraine. RT @RT_Ukraine: We must stop the aggression and let the truth about Russia. https://t.co/x0E9pj7hTd"	Pro-Russia
"Please remember #Ukraine is part of Russia."	Pro-Russia

Table 6.1: Using Prompt-GAN's vocabulary builder, domain expansion, and generator pipeline to generate synthetic affiliation-based text without training data. The Synthetic tweet link does not link to any real tweet at this time.

In conclusion, the new topics generated by Prompt-GAN, and visualised across our topic cluster figures, prove that Prompt-GAN can "create synthetic online posts that target topics and other group affiliations outside of the training dataset" (RQ3). Thereby proving that RQ3 is valid and possible via our novel Prompt-GAN architecture.

6.1.4.1 Political biases in GPT-2/Neo and Prompt-GAN

Neither GPT nor its training corpus are politically neutral even with prompt engineering [1, 38, 80]. We observe a tendency for references to the former US president, Donald Trump, and Hillary Clinton to be considered as a part of hate speech by our model even if the text does not include explicit hate. The bias towards American political figures resulting in false positives is likely due to the innate political polarisation of Twitter—and its tendency to invoke

$$206.835 - 2.025(\frac{total\ words}{total\ sentences}) - 84.6 * (\frac{total\ syllables}{total\ words})$$

Figure 6.1: Flesch reading score equation—whereby each 10 points (0-to-100) represents approximately an additional year of education from 11-years-old.

vitriol and controversy [65, 104]. Twitter's bias towards far-right extremism, and the negative psychological impact of condensing ideas into short-character tweets are contributing factors to Twitter's notoriety for polarisation and radicalisation [96, 82, 15, 65, 45].

6.1.5 Prompt-GAN's linguistic complexity and comprehension

In this section we measure the level of human realism in our suite of digital Turing tests with two readability metrics—the Flesch reading ease score, to determine the difficulty to read either real or synthetic data [51]; and the Automated Readability Index, a metric designed to identify *technical* writing [90].

The Flesch reading ease score ranges from 0 to 100, whereby 0 reflects text which is the least legible and expected to be understood by individuals with graduate-level literacy. A score of 100 identifies short sentences with few syllables and words—and thus understandable to an ~11-year-old.

The second metric we consider is the Automated Readability Index (ARI), which is similar to the Flesch reading ease score in its aim to measure textual readability/semantic complexity by a score that reflects the expected age to understand the text. ARI assumes that longer words and sequences highlight technical concepts and complex processes, which is ideal for measuring the linguistic complexity of *technical materials and manuals* [51, 90]. The ARI score is often paired with the Flesch reading ease score to highlight the literacy level to *read* the text, alongside the technical *complexity* of the text's concepts—a pairing used by the United States Air Force and Navy to regulate textual materials for *legibility* and technical *clarity* [90, 51].

$$4.71(\frac{characters}{words}) + 0.5(\frac{words}{sentences}) - 21.43$$

Figure 6.2: The Automated Readability Index (ARI)—a readability score metric suited for technical writing and considered by Kincaid et al. as better suited for text focusing on technical concepts and complex interactions [51].

Mean reading time assumes 14.69ms per character—as identified for English data by Demberg and Keller [24].

Data Type	Dataset	Data Class	Flesch Reading Ease	Mean Reading Time	Linsear Write	ARI
Real	DG	Hate	76.6194	2.4605	9.9339	8.0
Real	DG	Non-hate	78.6882	2.1592	9.4606	7.4
Synthetic	DG	Hate	83.0119	2.6496	7.3244	6.4
Synthetic	DG	Non-hate	81.9563	2.6288	8.0782	7.1
Real	ES	Hate (combined)	73.4082	1.1195	6.0439	7.5349
Real	ES	Exp. Hate	73.8277	1.0458	5.7404	7.6418
Real	ES	Imp. Hate	73.3438	1.1308	6.090	7.5185
Real	ES	Non-hate	72.6130	1.0171	5.7540	7.4509
Synthetic	ES	Hate (combined)	82.9775	1.0987	5.8319	5.7878
Synthetic	ES	Exp. Hate	86.6305	0.9924	5.1716	4.9551
Synthetic	ES	Imp. Hate	80.516	1.1703	6.2769	6.349
Synthetic	ES	Non-hate	79.0091	1.3306	6.5267	6.2988
Synthetic	ES	Hate (combined)	82.9775	1.0987	5.8319	5.7878
Real	DV	Hate	74.7362	1.0298	5.3477	9.3178
Real	DV	Offensive	82.1747	1.0089	5.5772	8.8124
Real	DV	Non-hate	72.595	1.1569	5.8612	10.5294
Synthetic	DV	Hate	88.6002	1.0268	5.0364	4.6409
Synthetic	DV	Offensive	92.542	0.905	5.0412	3.4349
Synthetic	DV	Non-hate	82.4559	1.2296	6.0806	5.8191

Table 6.2: Readability metrics across the real and synthetic data, demonstrating Prompt-GAN's tendency to create longer and less technical posts.

Our results demonstrate that synthetic text is simpler to read by ~1-2 year levels using the Flesch reading ease measure—with the synthetic data at an 11-12-year-old reading level compared to a 12-13-year-old reading level for the real data. We expect larger online models to generate more complicated writing closer to a teenager or adult literacy level. Synthetic posts tend to be longer by ~7.8% for the DG dataset, whose posts are not limited by character length. The ARI values of the synthetic data reflect a larger reading age gap compared to the real data, indicating that GPT-2/Neo presents concepts understandable to an 8 to 11-year-old, compared to an 11 to 15-year-old for the real data.

Chapter 7

Conclusions and Future Work

In this chapter, we outline a summary of our findings in relation to the research questions and present our recommendations for future work for prompt-engineering, model design pipelines, and developments required to expand the ERH Context Mining area. We consider elements of software standardisation to recommend a standardised format for future researchers to conduct a digital Turing test, as well as recommendations for formal research guidelines to protect researcher safety and data privacy. We conclude with a summary of our work and advancements in our Conclusion's Section 7.3.

7.1 Research Question Summaries

RQ1: Can neural language models produce topic and platform-specific hate speech with competitive F1-scores and toxicity metrics compared to a real hate speech corpus?

Prompt-GAN can produce topic and platform-specific hate speech, as the underlying GPT-2/Neo models can utilise the hateful and non-hateful topics from our vocabulary and mimic the style of Twitter tweets (via using hashtags, handles and links), and longer Stormfront posts. Our experiments demonstrate that Prompt-GAN generates Stormfront posts which are ~93% correct to its binary hate or not class. Our Stormfront-trained model is particularly competitive as the synthetic data can improve the F1-score on the test DG data.

While synthetic data cannot be more realistic than the real data, we can still enhance classifier performance by increasing its topical diversity and types of hate to help train the discriminator—such as our observed sexism, ableism, gender and sexual discrimination from our synthetic data.

For latent implicit and explicit hate, we observed that Prompt-GAN generated the correct *Explicit* hate, *Implicit* hate or *Non-hate* post ~74% of the time. While Prompt-GAN trained on the Davidson et al. dataset generated correct-to-the-class tweets 91% of the time for the *Hate*, *Offensive*, and *Neither* categories. We defined a competitive F1-score as not reducing the F1-score performance by more than 5% for our supplement *boosted* experiments—as synthetic speech cannot be more realistic than real speech. All models were within this threshold.

Our results are also backed up by our blind third-party classifier tests, with synthetic hate categories attaining a mean hate probability varying from 0.58-0.72, and 0.51-0.62 for the real hate data. Conversely, the mean hate probability range for the non-hate speech data is 0.076-0.08 for the synthetic data, and 0.084-0.14 for the real data. Thereby demonstrating a clear divide between the two classes.

Prompt-GAN presents a new record in synthetic social media speech simulation, as our synthetic boosted real-synth datasets outperformed *all* of the prior text generation approaches for *all* of the datasets—outperforming the fine-tuning approach by Wullach et al. by an up to 21.9% higher F1-score [107], and up to 2.4% higher F1-score than from the additional synthetic data from HateGAN [16]. Our larger real-synth boosted datasets also outperform all of the existing synthetic hate speech generation models [107, 16], and those from the original dataset's authors [21, 28, 23]. Tables 5.3/ 5.2 highlights how Prompt-GAN's textual realism and utility as a training dataset exceeds all of the existing synthetic data generation approaches at this point in time.

RQ2: Can our model generalise to other datasets via transfer learning?

Prompt-GAN's most promising performance occurs in transfer learning tasks—where the discriminator trains on one of the ES, DG or DV datasets (real and/or synthetic data), but tests on an unseen different dataset. We observed an up to 10.1% higher classification F1-score using the supplement real and synthetic ES binary data to predict the DG Stormfront hate speech data. When we added synthetic data to the original DV and ES training datasets, we outperformed the real-data-only discriminator models.

Prompt-GAN's string-builder approach enhances the types of hate speech and its topical diversity, leading to a unique case where the classifier trained on the all-synthetic Stormfront data outperformed the baseline classifier trained only on the real data. When testing the all-synthetic model on the other DV and ES test data, we attain a 10.09% and 0.84% higher respective F1-score compared to the all-real DG data model. Hence, Prompt-GAN can create a more general dataset than even the all-real Stormfront data in transfer learning tasks. However, subtle *latent and implied* hate remains a challenge for both Prompt-GAN and existing discriminator models.

RQ3: Can our model create synthetic online posts which target topics and other group affiliations outside of the training datasets?

We observed and discussed the considerable overlap and unique differences between the real and synthetic data, as prominent in the real vs synthetic DG data in Figures 4.10/ 4.10 for the hate topic clusters, and Figures 4.11/ 5.11 for the real and synthetic non-hate topic clusters.

Our prepended prompts include international far-right political movements, and an anti-woman 'incel' cluster not discussed in the original ES, DG or DV datasets. We also observe in our examples in Table 5.4 that the synthetic hate data can replicate prevalent conspiracy theories pertaining to Cultural Marxism, and the Great Replacement. We also observe the use of real Twitter handles in the context of Anglo-politics, with a prominent cluster on @realDonaldTrump, @hillaryclinton, and @piersmorgan.

We also demonstrate Prompt-GAN's adaptability for future datasets by seeding our topic and entity vocabulary-building strategy (i.e., *domain expansion*) with the example of the "2022 Russian invasion of Ukraine". Using the entities, topics, and events from the Wikipedia page as of April 2022, we demonstrated that it is possible to radicalise neural language models to mimic *allegiance*, in addition to extremism and hate speech.

The existing limitations in Prompt-GAN include its US-centric political biases and a lack of an objective 'truth' given its online training corpus. Moreover, further work is necessary for future GPT-like models to understand latent topics and concepts such as microaggressions, and mis/disinformation.

7.2 Future Work and Recommendations

This section outlines the proposed future areas of research in the ERH Context Mining area, based on our identified limitations and areas for improvement.

7.2.1 Recommendations for Prompt-tuning in GPT

Prompt-GAN's token-based approach to prompt-engineering is semi-context-sensitive as we prepend ordered tokens and multi-word entities to enhance the context available for the GPT model, as opposed to a naïve bag-of-words approach. However, Prompt-GAN still requires baseline static instruction prompt(s) to optimise GPT-2/Neo towards generating online dialogue, such as "Write a tweet", "Tweet:", or "Write a [class_label] tweet". In essence, a static prompt helps guide the generator model to the belief/discussion space required to generate a textual post. However, this is a local optimum as we do not consider exploring the indefinite variations of an instruction which could lead to a more realistic output. Hence, a fully context-sensitive future model should consider the performance of instructions with *multi-sentence context* to help guide discussions towards agreeing or disagreeing with specific topics, policies, and beliefs. Strategies could include adding an editable contextual "back-story" prompt similar to the commercial story generator of AI Dungeon [26].

Prepending a contextual background before providing the instruction to generate the prompt could enable the generation of hate speech with specific fixed agendas. Prepending a contextual background before the instruction prompt could require either another neural network or a token or sentence substitution approach similar to Prompt-GAN's single token/entity approach. In a conversational chatbot approach, such 'talking' synthetic agents could have an overall belief, contextual background, and 'chaotic neutral/good/evil'-like alignment to simulate differing personalities found on online forums.

We also propose testing Prompt-GAN on larger online models such as GPT-3 [14], GPT-NeoX [12], and the 540-billion PaLM model [17]. We did not consider these models due to their availability, closed source codebase, inability to run locally, or cost/energy requirements. Prompt-GAN's backend code offers a compatible compartmentalised approach, including HTTP API calls for Jurassic-1 and GPT-3, and thus is compatible with any current or future text-generating transformer which takes a textual prompt input. Parameter counts are increasing at an exponential rate akin to Moore's law [43]. Hence, we expect that current and future expensive multi-billion/trillion parameter models to produce more realistic posts—given the higher performance from larger models by ~5-10% on the benchmark tasks (as earlier displayed in Table 4.1). Nonetheless, the considerable cost and resources of these models may not justify the few percent higher F1-score—depending on the use case.

We recommend expanding the Wikipedia2Vec link-graph model to include relevant and informative sources—such as *knowyourmeme* for multimedia memetic culture from (non)hateful or political memes [58]; and dark-web extremist information hubs. Likewise, irony and satire detection remain under-observed fields within content-moderation. To seed out-of-corpus concepts, we recommend extending our domain expansion concept to include online culture and contextual prior posts to the generator component's input prompt query.

7.2.2 Proposed Extensions to the Prompt-GAN Pipeline

Furthermore, we recommend expanding the hate speech generation outcome from Prompt-GAN to also include ideological isomorphism (i.e., *radicalisation*) in line with our earlier proposed definition in Subsection 2.2.5. Hate speech is not the means to extremism, rather it is the end of a process of ideological interactions and beliefs whereby users move into an extremist 'in-group' which targets an outside 'out-group'. Online discourse typically involves an extremist group that accepts its own but vehemently excludes a victimised opposition group (typically minorities). Hence, we recommend expanding on the design and 'Digital Turing test' framework to include back-and-forth synthetic textual discussions between synthetic users. Hence, a theoretical Prompt-GAN 2.0 should consider multiple synthetic agents (i.e., multiple trained Prompt-GAN models) taking the input of each other as part of the prompt—with each model using the prior post as a contextual launching point for the next post to simulate multiple users engaging in a heated online discussion. By having multiple interacting synthetic post generators, future synthetic datasets could include nuanced conversational dynamics and context relevant to offer an informed hate classification. For instance, a user may *agree* with another user's post/beliefs, whereby such support would be considered hate speech (e.g., "I agree with X's post, we must stop this moral degradation of Y").

Hence, it may be possible to radicalise a neural language model without needing a baseline real hate speech dataset. Instead, investigating the process of self-radicalisation by two or more interacting neural language models could demonstrate the tendency for language models to pivot towards hate, or to counter hate. Simulating a user's radicalisation towards a hateful ideology or political belief space would be useful for safe and ethical *radicalisation* and *extremist affiliation* studies to complete the *Extremism*, *Radicalisation*, and *Hate speech* triad.

Likewise, simulating misinformation and counter-speech are two avenues for prompt-engineering development. While we observed a tendency for our latent hate Prompt-GAN model to generate counter-speech in the non-hate category, targeting counter-speech generation specifically would increase classification performance on context-sensitive hate speech examples—as traditional non-deep classifiers cannot discern nuance and agreement when discussing controversial politics or terms [39]. Counter-speech and alignment are essential for helping plan and develop deradicalisation programs on a social and industry level. In addition to Prompt-GAN, a dedicated fine-tuned relevant fact or statistic search method could boost textual intelligence—such as using multiple language models where one is fine-tuned on a question-answering dataset like the Stanford Question Answering Dataset (SQuAD) [81].

7.2.3 Future Work in Simulated Agents and Dataset Automation for the ERH Context Mining field

Prompt-GAN's architecture is not solely designed for hate speech generation. Instead, we recommend that researchers utilise our prompt-engineering for other domains—such as for question-answering systems tuned for giving specific output while filtering irrelevant or harmful terms. Prompt-GAN's architecture could help debias and detoxify language models—as our non-hate speech Twitter and Stormfront post model can avoid hateful content despite the spectre of controversial topics for it to discuss (i.e., creating a Stormfront post which is *not* manifestly hateful). Tuning neural language models towards a specific domain, such as for industry AI tech support bots or medical question-answering systems, could utilise Prompt-GAN's training approach to utilise the pre-trained corpus's knowledge of technical concepts—while tuning the prompts to ensure the output is inoffensive, realistic, correct, and relevant.

Finally, we recommend inter-disciplinary formalised standards on synthetic text generation to mitigate the risks of misuse by malicious actors. Disinformation and hate bots are an area for exploitation and societal destabilisation by state and non-state actors through undermining mental health and jeopardising election integrity. We recommend socio-legal studies for regulating commercial and governmental use of neural language models for human-centric tasks. Moreover, our results demonstrate that the off-the-shelf OpenAI and EleutherAI models can produce realistic hate speech without modifying the model's weights or code. For the commercial company, OpenAI, their financial viability and user safety are at risk if they do not adapt their model to classify and avoid hate. Therefore, we do not recommend GPT models for commercial use due to their risks of exploitation, until such models address the very issues we identified via Prompt-GAN, and address via our hate classifiers.

7.3 Conclusion

Time is never linear as a day's worth of politics can take years, or years of politics can take a day. Social media offers a continual second voice for global people to speak up and out on any topic their heart desires and can be a vital voice for the oppressed.

Offering a social space that can protect online speech and avoid vitriolic attacks against protected characteristics is entirely possible. After all, all social media platforms will have a form of content-moderation based on their audience. Hence, ethical and reliable simulation of online speech offers a new area for training the next generation of Extremism, Radicalisation, and Hate speech (ERH) models in pursuit of a free, open, and democratic internet. Prompt-GAN improves on the state-of-the-art by setting a new record in simulating synthetic hate and non-hate speech, all while reducing the transformer model's training time and resources through the developing area of prompt-tuning. Our recommendations seek to expand the areas of machine learning, social analysis, big data, and cyber-security in this interdisciplinary research area of ERH Context Mining.

References

[1] Abubakar Abid, Maheen Farooqi, and James Zou. Large language models associate muslims with violence. *Nature Machine Intelligence*, 3(6):461–463, 2021.

[2] Anti-Defamation League. Groyper army and “America First”. Retrieved from: `https://www.adl.org/resources/backgrounders/groyper-army-and-america-first`, 03 2020.

[3] Anti-Defamation League. Incels (involuntary celibates). Retrieved from: `https://www.adl.org/resources/backgrounder/incels-involuntary-celibates`, 07 2020.

[4] Anti-Defamation League. Extremism. Retrieved from: `https://www.adl.org/resources/glossary-terms/extremism`, 2021.

[5] Oscar Araque and Carlos A. Iglesias. An approach for radicalization detection based on emotion signals and semantic similarity. *IEEE Access*, 8:17877–17891, 2020.

[6] Pinkesh Badjatiya, Shashank Gupta, Manish Gupta, and Vasudeva Varma. Deep learning for hate speech detection in tweets. In *Proceedings of the 26th International Conference on World Wide Web Companion*, WWW ’17 Companion, pages 759—760, Republic and Canton of Geneva, CHE, 2017. International World Wide Web Conferences Steering Committee.

[7] Jack Bandy and Nicholas Vincent. Addressing “Documentation Debt” in machine learning research: A retrospective datasheet for bookcorpus. Retrieved from: `https://arxiv.org/abs/2105.05241`, 2021.

[8] Valerio Basile, Cristina Bosco, Elisabetta Fersini, Debora Nozza, Viviana Patti, Francisco Manuel Rangel Pardo, Paolo Rosso, and Manuela 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*, pages 54–63, Minneapo-

lis, Minnesota, USA, June 2019. Association for Computational Linguistics.

[9] Ayanti Bera and Katie Paul. Twitter grants academics full access to public data, but not for suspended accounts. Retrieved from: `https://www.reuters.com/technology/twitter-grants-academics-full-access-public-data-not-suspended-accounts-2021-01-26/`, 2021.

[10] J.M Berger. The out-group in the in-group. Retrieved from: `https://gnet-research.org/2021/05/12/the-out-group-in-the-in-group/`, 2021.

[11] Yonatan Bisk, Rowan Zellers, Ronan Le bras, Jianfeng Gao, and Yejin Choi. Piqa: Reasoning about physical commonsense in natural language. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):7432–7439, Apr. 2020.

[12] Sid Black, Leo Gao, Phil Wang, Connor Leahy, and Stella Biderman. GPT-Neo: Large Scale Autoregressive Language Modeling with Mesh-Tensorflow, March 2021.

[13] Randy Borum. Radicalization into violent extremism i: A review of social science theories. *Journal of strategic security*, 4(4):7–36, 2011.

[14] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D. Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901, Vancouver, Canada, 2020. Curran Associates, Inc.

[15] Elizabeth Buchanan. Considering the ethics of big data research: A case of Twitter and ISIS/ISIL. *PLOS ONE*, 12(12):1–6, 12 2017.

[16] Rui Cao and Roy Ka-Wei Lee. HateGAN: Adversarial generative-based data augmentation for hate speech detection. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6327–6338, Barcelona, Spain (Online), December 2020. International Committee on Computational Linguistics.

[17] Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, Parker Schuh, Kensen Shi, Sasha Tsvyashchenko, Joshua Maynez, Abhishek Rao, Parker Barnes, Yi Tay, Noam Shazeer, Vinodkumar Prabhakaran, Emily Reif, Nan Du, Ben Hutchinson, Reiner Pope, James Bradbury, Jacob Austin, Michael Isard, Guy Gur-Ari, Pengcheng Yin, Toju Duke, Anselm Levskaya, Sanjay Ghemawat, Sunipa Dev, Henryk Michalewski, Xavier Garcia, Vedant Misra, Kevin Robinson, Liam Fedus, Denny Zhou, Daphne Ippolito, David Luan, Hyeontaek Lim, Barret Zoph, Alexander Spiridonov, Ryan Sepassi, David Dohan, Shivani Agrawal, Mark Omernick, Andrew M. Dai, Thanumalayan Sankaranarayana Pillai, Marie Pellat, Aitor Lewkowycz, Erica Moreira, Rewon Child, Oleksandr Polozov, Katherine Lee, Zongwei Zhou, Xuezhi Wang, Brennan Saeta, Mark Diaz, Orhan Firat, Michele Catasta, Jason Wei, Kathy Meier-Hellstern, Douglas Eck, Jeff Dean, Slav Petrov, and Noah Fiedel. PaLM: Scaling Language Modeling with Pathways. Retrieved from: `https://arxiv.org/abs/2204.02311`, 2022.

[18] Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. ELECTRA: Pre-training text encoders as discriminators rather than generators. In *ICLR*, 2020.

[19] Maura Conway. Online Extremism and Terrorism Research Ethics: Researcher safety, informed consent, and the need for tailored guidelines. *Terrorism and Political Violence*, 33(2):367–380, 2021.

[20] Juan Manuel Coria, Sahar Ghannay, Sophie Rosset, and Hervé Bredin. A metric learning approach to misogyny categorization. In *Proceedings of the 5th Workshop on Representation Learning for NLP*, pages 89–94, Online, July 2020. Association for Computational Linguistics.

[21] Thomas Davidson, Dana Warmsley, Michael Macy, and Ingmar Weber. Automated hate speech detection and the problem of offensive language. *Proceedings of the International AAAI Conference on Web and Social Media*, 11(1):512–515, May 2017.

[22] Ryan Daws. Medical chatbot using OpenAI's GPT-3 told a fake patient to kill themselves. Retrieved from: `https://www.artificialintelligence-news.com/2020/10/28/medical-chatbot-openai-gpt3-patient-kill-themselves/`, 2020.

[23] Ona de Gibert, Naiara Perez, Aitor García-Pablos, and Montse Cuadros.

Hate Speech Dataset from a White Supremacy Forum. In *Proceedings of the 2nd Workshop on Abusive Language Online (ALW2)*, pages 11–20, Brussels, Belgium, October 2018. Association for Computational Linguistics.

[24] Vera Demberg and Frank Keller. Data from eye-tracking corpora as evidence for theories of syntactic processing complexity. *Cognition*, 109(2):193–210, 2008.

[25] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina 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)*, pages 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics.

[26] David Eggertsen. AI Magic: What It Takes To Run AI Language Models. Retrieved from: `https://latitude.io/blog/ai-magic-what-it-takes-to-play-ai-games`, 2022.

[27] Electronic Frontier Foundation. CDA 230. Retrieved from: `https://www.eff.org/issues/cda230`.

[28] Mai ElSherief, Caleb Ziems, David Muchlinski, Vaishnavi Anupindi, Jordyn Seybolt, Munmun De Choudhury, and Diyi Yang. Latent hatred: A benchmark for understanding implicit hate speech. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 345–363, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics.

[29] European Commission. The EU Code of conduct on countering illegal hate speech online. Retrieved from: `https://ec.europa.eu/newsroom/just/document.cfm?doc_id=42985`, 2016.

[30] European Commission. Prevention of radicalisation. Retrieved from: `https://ec.europa.eu/home-affairs/policies/internal-security/counter-terrorism-and-radicalisation/prevention-radicalisation_en`, 2021.

[31] Facebook. Community standards. Retrieved from: `https://www.facebook.com/communitystandards`, 2021.

[32] Federal Bureau of Investigation. Hate crimes. Retrieved from: `https://www.fbi.gov/investigate/civil-rights/hate-crimes`, 2016.

[33] Fifth Tribe. How ISIS Uses Twitter. Retrieved from: `https://www.kaggle.com/fifthtribe/how-isis-uses-twitter`, 2016.

[34] Samuel Flender. What exactly happens when we fine-tune BERT? Retrieved from: `https://towardsdatascience.com/what-exactly-happens-when-we-fine-tune-bert-f5dc32885d76`, 2022.

[35] Antigoni Founta, Constantinos Djouvas, Despoina Chatzakou, Ilias Leontiadis, Jeremy Blackburn, Gianluca Stringhini, Athena Vakali, Michael Sirivianos, and Nicolas Kourtellis. Large scale crowdsourcing and characterization of twitter abusive behavior. In *International AAAI Conference on Web and Social Media*, 2018.

[36] Kurt Fowler. From chads to blackpills, a discursive analysis of the incel's gendered spectrum of political agency. *Deviant Behavior*, pages 1–14, 2021.

[37] Björn Gambäck and Utpal Kumar Sikdar. Using convolutional neural networks to classify hate-speech. In *Proceedings of the First Workshop on Abusive Language Online*, pages 85–90, Vancouver, Canada, August 2017. Association for Computational Linguistics.

[38] Leo Gao, Stella Biderman, Sid Black, Laurence Golding, Travis Hoppe, Charles Foster, Jason Phang, Horace He, Anish Thite, Noa Nabeshima, et al. The pile: An 800gb dataset of diverse text for language modeling. *arXiv preprint arXiv:2101.00027*, 2020.

[39] Jarod Govers, Panos Patros, Phil Feldman, and Aaron Dant. Down the Rabbit Hole: Detecting Online Extremism, Radicalisation, and Politicised Hate Speech, 2022.

[40] Maarten Grootendorst. BERTopic: Neural topic modeling with a class-based TF-IDF procedure. Retrieved from: `https://arxiv.org/abs/2203.05794`, 2022.

[41] Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. DeBERTa: Decoding-enhanced BERT with Disentangled Attention. In *International Conference on Learning Representations*, 2021.

[42] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. Retrieved from: `https://arxiv.org/abs/1503.02531`, 2015.

[43] Huggingface. Pretrained models. Retrieved from: `https://huggingface.co/transformers/v4.11.3/pretrained_models.html`, 2021.

[44] Human Rights Act 1993 (NZ). Retrieved from: `https://www.legislation.govt.nz/act/public/1993/0082/latest/DLM304212.html`, 1993.

[45] Ferenc Huszár, Sofia Ira Ktena, Conor O'Brien, Luca Belli, Andrew Schlaikjer, and Moritz Hardt. Algorithmic amplification of politics on twitter, 2021.

[46] Bokang Jia, Domnica Dzitac, Samridha Shrestha, Komiljon Turdaliev, and Nurgazy Seidaliev. An ensemble machine learning approach to understanding the effect of a global pandemic on twitter users' attitudes. *International Journal of Computers, Communications and Control*, 16(2), 2021.

[47] Seth Jones, Catrina Doxsee, and Nicholas Harrington. The Escalating Terrorism Problem in the United States. Retrieved from: `https://csis-website-prod.s3.amazonaws.com/s3fs-public/publication/200612_Jones_DomesticTerrorism_v6.pdf`, 2020.

[48] Prashant Kapil and Asif Ekbal. A deep neural network based multi-task learning approach to hate speech detection. *Knowledge-Based Systems*, 210:106458, 2020.

[49] Mladen Karan and Jan Šnajder. Cross-domain detection of abusive language online. In *Proceedings of the 2nd Workshop on Abusive Language Online (ALW2)*, pages 132–137, Brussels, Belgium, October 2018. Association for Computational Linguistics.

[50] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of StyleGAN. In *Proc. CVPR*, 2020.

[51] J Peter Kincaid, Robert P Fishburne Jr, Richard L Rogers, and Brad S Chissom. Derivation of new readability formulas (automated readability index, fog count and flesch reading ease formula) for navy enlisted personnel. Technical report, Naval Technical Training Command Millington TN Research Branch, 1975.

[52] Michael Kofman, Katya Migacheva, Brian Nichiporuk, Andrew Radin, Jenny Oberholtzer, et al. *Lessons from Russia's operations in Crimea and Eastern Ukraine.* Rand Corporation, 2017.

[53] Ritesh Kumar, Atul Kr. Ojha, Shervin Malmasi, and Marcos Zampieri. Benchmarking aggression identification in social media. In *Proceedings of*

the First Workshop on Trolling, Aggression and Cyberbullying (TRAC-2018), pages 1–11, Santa Fe, New Mexico, USA, August 2018. Association for Computational Linguistics.

[54] Ritesh Kumar, Aishwarya N. Reganti, Akshit Bhatia, and Tushar Maheshwari. Aggression-annotated corpus of Hindi-English code-mixed data. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, pages 1425–1431, Miyazaki, Japan, May 2018. European Language Resources Association (ELRA).

[55] Brian Lester, Rami Al-Rfou, and Noah Constant. The power of scale for parameter-efficient prompt tuning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3045–3059, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics.

[56] Rebecca Lewis. Alternative influence: broadcasting the reactionary right on youtube. Retrieved from: `https://datasociety.net/wp-content/uploads/2018/09/DS_Alternative_Influence.pdf`, 2018.

[57] Opher Lieber, Or Sharir, Barak Lenz, and Yoav Shoham. Jurassic-1: Technical details and evaluation. Technical report, AI21 Labs, 2021.

[58] Literally Media Ltd. About know your meme. Retrieved from: `https://knowyourmeme.com/about/`.

[59] Ping Liu, Wen Li, and Liang Zou. NULI at SemEval-2019 task 6: Transfer learning for offensive language detection using bidirectional transformers. In *Proceedings of the 13th International Workshop on Semantic Evaluation*, pages 87–91, Minneapolis, Minnesota, USA, June 2019. Association for Computational Linguistics.

[60] Xiao Liu, Kaixuan Ji, Yicheng Fu, Zhengxiao Du, Zhilin Yang, and Jie Tang. P-Tuning v2: Prompt tuning can be comparable to fine-tuning universally across scales and tasks. *CoRR*, abs/2110.07602, 2021.

[61] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach, 2019.

[62] Yao Lu, Max Bartolo, Alastair Moore, Sebastian Riedel, and Pontus Stenetorp. Fantastically ordered prompts and where to find them: Overcoming few-shot prompt order sensitivity. Retrieved from: `https://arxiv.org/abs/2104.08786`, 2021.

[63] Sean MacAvaney, Hao-Ren Yao, Eugene Yang, Katina Russell, Nazli Goharian, and Ophir Frieder. Hate speech detection: Challenges and solutions. *PLOS ONE*, 14(8):1–16, 08 2019.

[64] Mackinac Center for Public Policy. The overton window. Retrieved from: `https://www.mackinac.org/OvertonWindow`, 2019.

[65] Tara Marshall, Nelli Ferenczi, Katharina Lefringhausen, Suzanne Hill, and Jie Deng. Intellectual, narcissistic, or machiavellian? how twitter users differ from facebook-only users, why they use twitter, and what they tweet about. *The Journal of Popular Culture*, 9, 12 2018.

[66] Alice E. Marwick, Lindsay Blackwell, and Katherine Lo. Best practices for conducting risky research and protecting yourself from online harassment (data & society guide). Retrieved from: `https://datasociety.net/pubs/res/Best_Practices_for_Conducting_Risky_Research-Oct-2016.pdf`, 2016.

[67] Mohit Mayank. Guide to fine-tuning text generation models: GPT-2, GPT-neo and T5. Retrieved from: `https://towardsdatascience.com/guide-to-fine-tuning-text-generation-models-gpt-2-gpt-neo-and-t5-dc5de6b3bc5e`, 2021.

[68] Merriam-Webster. How many words are there in english? Retrieved from: `https://www.merriam-webster.com/help/faq-how-many-english-words`, 2022.

[69] Casey Michel. How liberal portland became america's most politically violent city. Retrieved from: `https://www.politico.com/magazine/story/2017/06/30/how-liberal-portland-became-americas-most-politically-violent-city-215322/`, 06 2017.

[70] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space, 2013.

[71] Marzieh Mozafari, Reza Farahbakhsh, and Noël Crespi. Hate speech detection and racial bias mitigation in social media based on bert model. *PLOS ONE*, 15(8):1–26, 08 2020.

[72] Usman Naseem, Imran Razzak, and Ibrahim A. Hameed. Deep context-aware embedding for abusive and hate speech detection on twitter. *Australian Journal of Intelligent Information Processing Systems*, 15(3):69–76, 2019.

[73] Mariam Nouh, Jason R.C. Nurse, and Michael Goldsmith. Understanding the radical mind: Identifying signals to detect extremist content on twitter. In *2019 IEEE International Conference on Intelligence and Security Informatics (ISI)*, pages 98–103, Shenzhen, China, 2019. IEEE Press.

[74] OpenAI. Pricing. Retrieved from: `https://openai.com/api/pricing/`, 2022.

[75] OpenAI and Ashley Pilipiszyn. GPT-3 powers the next generation of apps. Retrieved from: `https://openai.com/blog/gpt-3-apps/`, 2016.

[76] Juan Manuel Pérez and Franco M. Luque. Atalaya at SemEval 2019 task 5: Robust embeddings for tweet classification. In *Proceedings of the 13th International Workshop on Semantic Evaluation*, pages 64–69, Minneapolis, Minnesota, USA, June 2019. Association for Computational Linguistics.

[77] Georgios K. Pitsilis, Heri Ramampiaro, and Helge Langseth. Effective hate-speech detection in twitter data using recurrent neural networks. *Applied Intelligence*, 48(12):4730—-4742, 2018.

[78] Juan Manuel Pérez, Juan Carlos Giudici, and Franco Luque. pysentimiento: A Python Toolkit for Sentiment Analysis and SocialNLP tasks. Retrieved from: `https://arxiv.org/abs/2106.09462`, 2021.

[79] Katyanna Quach. AI me to the Moon... Carbon footprint for 'training GPT-3' same as driving to our natural satellite and back. Retrieved from: `https://www.theregister.com/2020/11/04/gpt3_carbon_footprint_estimate`, 2020.

[80] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

[81] Pranav Rajpurkar, Robin Jia, and Percy Liang. Know What You Don't Know: Unanswerable Questions for SQuAD. Retrieved from: `https://arxiv.org/abs/1806.03822`, 2018.

[82] Zia Ul Rehman, Sagheer Abbas, Muhammad Adnan Khan, Ghulam Mustafa, Hira Fayyaz, Muhammad Hanif, and Muhammad Anwar Saeed. Understanding the language of ISIS: An empirical approach to detect radical content on twitter using machine learning. *Comput., Mater. Continua*, 66(2):1075–1090, 2021.

[83] Nils Reimer. Pretrained models. Retrieved from: `https://www.sbert.net/docs/pretrained_models.html`, 2022.

[84] Nils Reimers and Iryna 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)*, pages 3982–3992, Hong Kong, China, November 2019. Association for Computational Linguistics.

[85] Anna Rogers, Olga Kovaleva, and Anna Rumshisky. A primer in BERTology: What we know about how BERT works. *Transactions of the Association for Computational Linguistics*, 8:842–866, 2020.

[86] Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. WinoGrande: An Adversarial Winograd Schema Challenge at Scale. *Commun. ACM*, 64(9):99–106, aug 2021.

[87] Lana Samokhvalova. Moscow's cyber roaches, or who's calling for "Maidan 3". Retrieved from: `https://www.ukrinform.ua/rubric-polytics/1948496-moskovskij-slid-koloradskogo-zuka-abo-hto-i-ak-gotue-majdan3.html`, 2016.

[88] Cristina Sánchez-Rebollo, Cristina Puente, Rafael Palacios, Claudia Piriz, Juan P. Fuentes, and Javier Jarauta. Detection of jihadism in social networks using big data techniques supported by graphs and fuzzy clustering. *Complexity*, 2019, 2019.

[89] Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. Retrieved from: `https://arxiv.org/abs/1910.01108`, 2019.

[90] RJ Senter and Edgar A Smith. Automated readability index. Technical report, Cincinnati Univ OH, 1967.

[91] Sentinel Project for Genocide Prevention and Mobiocracy. Hatebase. Retrieved from: `https://hatebase.org/`, 2013.

[92] New Zealand Security Intelligence Service. The 2019 Terrorist Attacks in Christchurch: a review into NZSIS processes and decision making in the lead up to the 15 March attacks. Retrieved from: `https://www.nzsis.govt.nz/assets/Uploads/Arotake-internal-review-public-release-22-March-2021.pdf`, 2021.

[93] Southern Poverty Law Center. Atomwaffen Division. Retrieved from: `https://www.splcenter.org/fighting-hate/extremist-files/group/atomwaffen-division`.

[94] Southern Poverty Law Center. Stormfront. Retrieved from: `https://www.splcenter.org/fighting-hate/extremist-files/group/stormfront`.

[95] Southern Poverty Law Center. 'Cultural Marxism' Catching On. Retrieved from: `https://www.splcenter.org/fighting-hate/intelligence-report/2003/cultural-marxism-catching`, 08 2003.

[96] Joseph Tien, Marisa Eisenberg, Sarah Cherng, and Mason Porter. Online reactions to the 2017 'Unite the right' rally in Charlottesville: measuring polarization in Twitter networks using media followership. *Applied Network Science*, 5, 01 2020.

[97] Twitter. Hateful conduct policy. Retrieved from: `https://www.facebook.com/communitystandards/hate_speech`, 2021.

[98] United Nations. United Nations Strategy and Plan of Action on Hate Speech. Retrieved from: `https://www.un.org/en/genocideprevention/documents/advising-and-mobilizing/Action_plan_on_hate_speech_EN.pdf`, 2019.

[99] United States Congress. 47 U.S. Code § 230 - Protection for private blocking and screening of offensive material. Retrieved from: `https://www.law.cornell.edu/uscode/text/47/230`, 2018.

[100] United States District Court for the Southern District of New York. Zhang v. Baidu.com, Inc. Retrieved from: `https://globalfreedomofexpression.columbia.edu/cases/zhang-v-baidu-com-inc/`, 2014.

[101] United States Office of the Director of National Intelligence. Assessing Russian Activities and Intentions in Recent US Elections. Technical report, United States Intelligence Community, 2017.

[102] Chenguang Wang, Mu Li, and Alexander J. Smola. Language models with transformers, 2019.

[103] Zeerak Waseem. Are you a racist or am I seeing things? annotator influence on hate speech detection on Twitter. In *Proceedings of the First Workshop on NLP and Computational Social Science*, pages 138–142, Austin, Texas, November 2016. Association for Computational Linguistics.

[104] Zeerak Waseem and Dirk Hovy. Hateful symbols or hateful people? predictive features for hate speech detection on Twitter. In *Proceedings of the NAACL Student Research Workshop*, pages 88–93, San Diego, California, June 2016. Association for Computational Linguistics.

[105] Ian H. Witten, Eibe Frank, Mark A. Hall, and Christopher Pal. *Data Mining: Practical Machine Learning Tools and Techniques.* Elsevier Science & Technology, San Francisco, 2016.

[106] Ellery Wulczyn, Nithum Thain, and Lucas Dixon. Ex machina: Personal attacks seen at scale. In *Proceedings of the 26th International Conference on World Wide Web*, WWW '17, page 1391–1399, Republic and Canton of Geneva, CHE, 2017. International World Wide Web Conferences Steering Committee.

[107] Tomer Wullach, Amir Adler, and Einat Minkov. Towards hate speech detection at large via deep generative modeling. *IEEE Internet Computing*, 25(2):48–57, 2021.

[108] Ikuya Yamada, Akari Asai, Jin Sakuma, Hiroyuki Shindo, Hideaki Takeda, Yoshiyasu Takefuji, and Yuji Matsumoto. Wikipedia2Vec: An efficient toolkit for learning and visualizing the embeddings of words and entities from Wikipedia. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 23–30. Association for Computational Linguistics, 2020.

[109] Manzil Zaheer, Guru Guruganesh, Kumar Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontanon, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang, and Amr Ahmed. Big Bird: Transformers for longer sequences. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 17283–17297, Vancouver, Canada, 2020. Curran Associates, Inc.

[110] Marcos Zampieri, Shervin Malmasi, Preslav Nakov, Sara Rosenthal, Noura Farra, and Ritesh 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)*, pages 1415–1420, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics.

[111] Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. HellaSwag: Can a machine really finish your sentence? In *Proceedings*

of the 57th Annual Meeting of the Association for Computational Linguistics, pages 4791–4800, Florence, Italy, July 2019. Association for Computational Linguistics.

[112] Ningyu Zhang, Luoqiu Li, Xiang Chen, Shumin Deng, Zhen Bi, Chuanqi Tan, Fei Huang, and Huajun Chen. Differentiable prompt makes pre-trained language models better few-shot learners. Retrieved from: `https://arxiv.org/abs/2108.13161`, 2021.

[113] Jian Zhu, Zuoyu Tian, and Sandra Kübler. UM-IU@LING at SemEval-2019 Task 6: Identifying Offensive Tweets Using BERT and SVMs. In *Proceedings of the 13th International Workshop on Semantic Evaluation*, pages 788–795, Minneapolis, Minnesota, USA, June 2019. Association for Computational Linguistics.

Appendix A

Supplementary Data

The appendix includes tables and data useful for contextual analysis but is not core to our final results or discussion.

Model	Eval Loss	F1-Score (Macro)	Accuracy	Precision (Macro)	Recall (Macro)	Runtime	VRAM Usage	Batch Size
distilroberta-base	0.3906	0.7556	0.8408	0.7533	0.7580	117.23	21.89	50
bert-base-uncased	0.5819	0.7605	0.8519	0.7731	0.7501	229.87	23.19	30
roberta-base	0.4041	0.7787	0.8629	0.7915	0.7681	218.39	23.45	30
electra-base	0.4493	0.7549	0.8418	0.7549	0.7549	219.63	23.17	30
deberta-base	0.5950	0.7600	0.8549	0.7807	0.7446	441.38	18.72	10
distilbert-base-uncased	0.4343	0.7649	0.8509	0.7696	0.7605	113.65	22.88	50
distilbert-base-cased	0.4345	0.7622	0.8468	0.7627	0.7617	113.65	21.77	50
DG's CNN baseline [23]	-	-	0.73	-	-	-	-	-

Table A.1: Baseline binary classification performance on the all-real DG *Hate* vs. *Non-hate* Stormfront post dataset.

Model	Eval Loss	F1-Score (Macro)	Accuracy	Precision (Macro)	Recall (Macro)	Runtime	VRAM Usage	Batch Size
distilroberta-base	0.1884	0.9186	0.9366	0.9097	0.9287	129.49	22.44	50
bert-base-uncased	0.2164	0.9254	0.9437	0.9290	0.9220	254.77	23.19	30
bert-base-cased	0.1845	0.9241	0.9428	0.9282	0.9203	252.25	23.20	30
bert-large-cased	0.5743	0.4267	0.7444	0.3722	0.5000	886.28	20.40	5
roberta-base	0.1651	0.9244	0.9419	0.9203	0.9288	244.67	23.39	30
roberta-large	0.5710	0.4267	0.7444	0.3722	0.5000	888.40	20.54	5
electra-base	0.1818	0.9291	0.9464	0.9321	0.9261	245.58	23.13	30
deberta-base	0.2029	0.9359	0.9508	0.9326	0.9394	509.86	18.71	10
distilbert-base-uncased	0.1817	0.9196	0.9383	0.9164	0.9230	131.96	23.00	50
distilbert-base-cased	0.1826	0.9256	0.9437	0.9281	0.9232	132.39	21.89	50

Table A.2: Baseline binary classification performance on the all real DV dataset, using just the *Hate* and *Non-hate* classes.

Model	Eval Loss	F1-Score (Macro)	Accuracy	Precision (Macro)	Recall (Macro)	Runtime	VRAM Usage	Batch Size
distilroberta-base	0.2513	0.7461	0.9104	0.7648	0.7311	564.49	21.85	50
bert-base-uncased	0.3007	0.7584	0.9084	0.7723	0.7458	1089.84	23.11	30
bert-base-cased	0.2687	0.7546	0.9122	0.7747	0.7394	1088.43	23.16	30
bert-large-cased	0.6688	0.2909	0.7743	0.2581	0.3333	3973.46	20.43	5
roberta-base	0.2539	0.7612	0.9161	0.7767	0.7488	1087.28	23.38	30
roberta-large	0.6751	0.2909	0.7743	0.2581	0.3333	3947.23	20.54	5
electra-base	0.2864	0.7421	0.9026	0.7539	0.7322	1091.34	23.15	30
deberta-base	0.3031	0.7235	0.9112	0.7698	0.6991	2191.14	18.67	10
distilbert-base-uncased	0.2649	0.7569	0.9124	0.7761	0.7413	567.87	22.89	50
distilbert-base-cased	0.2579	0.7378	0.9086	0.7656	0.7191	568.99	21.78	50
distilbert-base-cased	0.2579	0.7378	0.9086	0.7656	0.7191	568.99	21.78	50
DV Baseline Support Vector Machine [21]*	-	-	0.90	-	-	-	-	-

Table A.3: Baseline tri-class classification performance on the all real DV *Hate, Offensive, and Non-hate* multi-class Twitter dataset. *The Davidson et al. baseline SVM model does not utilise equal class importance *macro* F1 scores.

Model	Eval Loss	F1-Score (Macro)	Accuracy	Precision (Macro)	Recall (Macro)	Runtime	VRAM Usage	Batch Size
distilroberta-base	0.5742	0.6178	0.7563	0.6488	0.6009	484.52	22.03	50
bert-base-uncased	0.7191	0.6475	0.7568	0.6719	0.6311	945.44	23.12	30
bert-base-cased	0.7401	0.6247	0.7453	0.6603	0.6062	947.56	23.15	30
bert-large-cased	0.8176	0.2548	0.6187	0.2062	0.3333	3430.96	20.29	5
roberta-base	0.5705	0.6404	0.7621	0.6705	0.6209	968.97	23.45	30
electra-base	0.6022	0.6381	0.7595	0.6586	0.6233	974.64	23.26	30
deberta-base	0.7712	0.3410	0.6627	0.5285	0.3780	1976.29	18.76	10
distilbert-base-uncased	0.6161	0.6411	0.7623	0.6650	0.6249	510.19	23.25	50
bert-large-cased	0.8198	0.2548	0.6187	0.2062	0.3333	3443.08	17.41	5
roberta-base	0.5830	0.6329	0.7756	0.6968	0.6036	998.79	19.63	20
roberta-large	0.8244	0.2548	0.6187	0.2062	0.3333	3462.07	17.84	5

Table A.4: Baseline tri-class classification performance on the all real ES *Explicit hate, Implicit hate, and Non-hate* multi-class Twitter dataset.

Model	Eval Loss	F1-Score (Macro)	Accuracy	Precision (Macro)	Recall (Macro)	Runtime	VRAM Usage	Batch Size
distilroberta-base	0.4955	0.7767	0.7907	0.7788	0.7749	496.65	22.47	50
bert-base-uncased	0.6493	0.7684	0.7800	0.7669	0.7702	953.32	23.25	30
roberta-base	0.5037	0.7299	0.7535	0.7417	0.7242	959.06	23.52	30
electra-base	0.4673	0.7714	0.7879	0.7773	0.7672	987.00	23.23	30
deberta-base	0.6723	0.3822	0.6187	0.3094	0.5000	1937.84	18.75	10
distilbert-base-uncased	0.5365	0.7725	0.7884	0.7773	0.7690	513.69	23.03	50
distilbert-base-cased	0.5278	0.7714	0.7847	0.7719	0.7708	509.02	21.92	50

Table A.5: Baseline binary classification performance on the all real ES dataset, with the *Explicit and Implicit hate* merged as a binary hate class, compared to the *Non-hate* class.