Hate Speech Diffusion in Twitter Social Media

BY

SAKSHI MAKKAR

MT18013

MTech in CSE (Specialisation in AI)

INDRAPRASTHA INSTITUTE *of*
INFORMATION TECHNOLOGY
DELHI

Supervisor:
Dr Tanmoy Chakraborty
Assistant Professor IIIT-Delhi

A thesis submitted in partial fulfilment of
the requirements for the degree of
MTech (Specialisation in Artificial Intelligence)

Department of Computer Science and Engineering
Indraprastha Institute of Information Technology
Delhi, India

July, 2020

Certificate

This is to certify that the thesis titled **"Hate speech diffusion in Twitter Social Media"** submitted by **Sakshi Makkar** for the partial fulfillment of the requirements for the degree of *Master of Technology* in *Computer Science & Engineering* is a record of the bonafide work carried out by her under my guidance and supervision at Indraprastha Institute of Information Technology, Delhi. In my opinion, the thesis has reached the standards fulfilling the requirements of the regulations relating to the degree. This work has not been submitted anywhere else for the reward of any other degree.

June, 2020

Dr Tanmoy Chakraborty
Dept. of Computer Science Engineering
Indraprastha Institute of Information Technology Delhi
New Delhi 110020

Acknowledgements

I would like to thank my advisor Dr. Tanmoy Chakraborty at IIIT Delhi for providing me the opportunity to work on this thesis which involves real world application. I would also like to provide him my sincere gratitude for his valuable guidance, suggestions, continuous support and encouragement. A big thanks to him for helping me walk through the new avenues of research papers and publications.

I would like to thank Subhabhrata Dutta and Sarah Masud, for their continous guidance and efforts in this Thesis.

I would also like to thank thesis committee members for evaluating my work. I would like to thank my friends and college mates for their immense support.

Most importantly, none of this would have happened without the love and patience of my family - my parents, to whom this thesis is dedicated. I would like to express my heart-felt gratitude to my family.

Author

Sakshi Makkar

Abstract

Online hate speech, particularly over microblogging platforms like Twitter, has emerged as arguably the most severe issue of the past decade. Several countries have reported a steep rise in hate crimes infuriated by malicious hate campaigns. While the detection of hate speech is one of the emerging research areas, the generation and spread of topic-dependent hate in the information network remains underexplored. In this work, we focus on exploring user behavior, which triggers the genesis of hate speech on Twitter and how it diffuse via retweets. We crawl a large-scale dataset of tweets, retweets, user activity history, and follower networks, comprising over 161 million tweets from more than 41 million unique users. We also collect over $600k$ contemporary news articles published online. We characterize different signals of information that govern these dynamics. Our analyses differentiate the diffusion dynamics in the presence of hate from usual information diffusion. This motivates us to formulate the modeling problem in a *topic-aware setting* with real-world knowledge. For predicting the initiation of hate speech for any given hashtag, we propose multiple feature-rich models, with the best performing one achieving a macro F1 score of 0.65. Meanwhile, to predict the retweet dynamics on Twitter, we propose `RETINA`, a novel neural architecture that incorporates exogenous influence using scaled dot-product attention. `RETINA` achieves a macro F1 score of 0.85, outperforming multiple state-of-the-art models. Our analysis reveals the superlative power of `RETINA` to predict the retweet dynamics of hateful content compared to the existing diffusion models.

Contents

List of Figures

List of Tables

CHAPTER 1

Introduction

The Internet is one of the greatest boons to society since it has brought together people from different race, religion, and nationality. Various social media websites like Twitter and Facebook have connected billions of people and allowed them to share their ideas and opinions instantly. They act as a platform for users to exchange their views or gain knowledge about various trends in the society. Due to the immense popularity of these social media platforms they continue to be one of the most preferred sources of information dissemination in human society. Some people however, misuse this medium to spread hateful or offensive contents.

1.1 Basic Terminologies

Here are few basic terminologies that need to be known before proceeding further since these terminologies are used throughout the thesis.

1.1.1 Twitter

According to Wikipedia, "Twitter (/twtr/)[1] is an American microblogging and social networking service on which users post and interact with messages known as "tweets". Tweets can be of 280 characters or less in length. Twitter can be used for various forms of communication.

[1] https://en.wikipedia.org/wiki/Twitter

1.1.2 Tweet and Retweet

The message posted on twitter is known as tweet while a Retweet is a repost of message or a tweet of another user. It begins with characters 'RT'.

1.1.3 Followers and Followees

The followers of a user are the one's who will receive tweets posted by the user. Followers can further Retweet, like or comment on the tweet. Followees on the other hand are the one's that user follows.

1.2 Hate Speech

Twitter updated its "Hateful Conduct Policy" in 2017 and defines hate speech as any tweet that 'promotes violence against other people on the basis of race, ethnicity, national origin, sexual orientation, gender, gender identity, religious affiliation, age, disability, or serious disease' [2]. This definition includes tweets that may appeal violence against some disadvantaged group.

Many hate crimes against minority and backward communities have been directly linked with hateful campaigns circulated over Facebook, Twitter, Gab, and many other online platforms [1, 2]. Online social media has provided an unforeseen speed of information spread, aided by the fact that the power of content generation is handed to every user of these platforms. Extremists have exploited this phenomenon to disseminate hate campaigns to a degree where manual monitoring is too costly, if not impossible.

Thankfully, the research community has been observing a spike of works related to online hate speech, with a vast majority of them focusing on the problem of automatic detection of hate from online text [3]. However, as Ross et al. [4] pointed it out, even manual identification of hate speech comes with ambiguity due to the differences in the definition of hate. Also, an important signal of hate speech is the presence of specific words/phrases, which vary

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

significantly across topics/domains. Tracking such a diverse socio-linguistic phenomenon in real-time is impossible for automated, large-scale platforms.

An alternative approach can be to track potential groups of users who have a history of spreading hate. As Matthew et al. [5] suggested, such users are often a very small fraction of the total users but generate a sizeable portion of the content. Moreover, the severity of hate speech lies in the degree of its spread, and an early prediction of the diffusion dynamics may help to combat online hate speech to a new extent altogether. However, a tiny fraction of the existing literature seeks to explore the problem quantitatively. Matthew et al. [5] put up an insightful foundation for this problem by analyzing the dynamics of hate diffusion in Gab[3]. However, they do not tackle the problem of modeling the diffusion and restrict themselves to identifying the different characteristics of hate speech in Gab.

Hate speech on Twitter Twitter, as one of the largest micro-blogging platforms with a worldwide user base, has a long history of accommodating hate speech, cyberbullying, and toxic behavior. Recently, it has come hard at such contents multiple times[4][5], and a certain fraction of hateful tweets are often removed upon identification. However, a large majority of such tweets still circumvent Twitter's filtering. In this work, we choose to focus on the dynamics of hate speech on Twitter mainly due to two reasons: (i) the wide-spread usage of Twitter compared to other platforms provides scope to grasp the hate diffusion dynamics in a more realistic manifestation, and (ii) understanding how hate speech emerges and spreads even in the presence of some top-down checking measures, compared to unmoderated platforms like Gab.

Diffusion patterns of hate vs. non-hate on Twitter. Hate speech is often characterized by the formation of *echo-chambers*, i.e., only a small group of people engaging with such contents repeatedly. In Figure 1.1, we compare the temporal diffusion dynamics of hateful vs. non-hate tweets (see Sections 3.1 and 5.0.1 for the details of our dataset and hate detection methods, respectively). Following the standard information diffusion terminology, the set of

[3] https://gab.com/

[4] https://www.bbc.com/news/technology-42376546

[5] https://blog.twitter.com/en_us/topics/company/2019/hatefulconductupdate.html

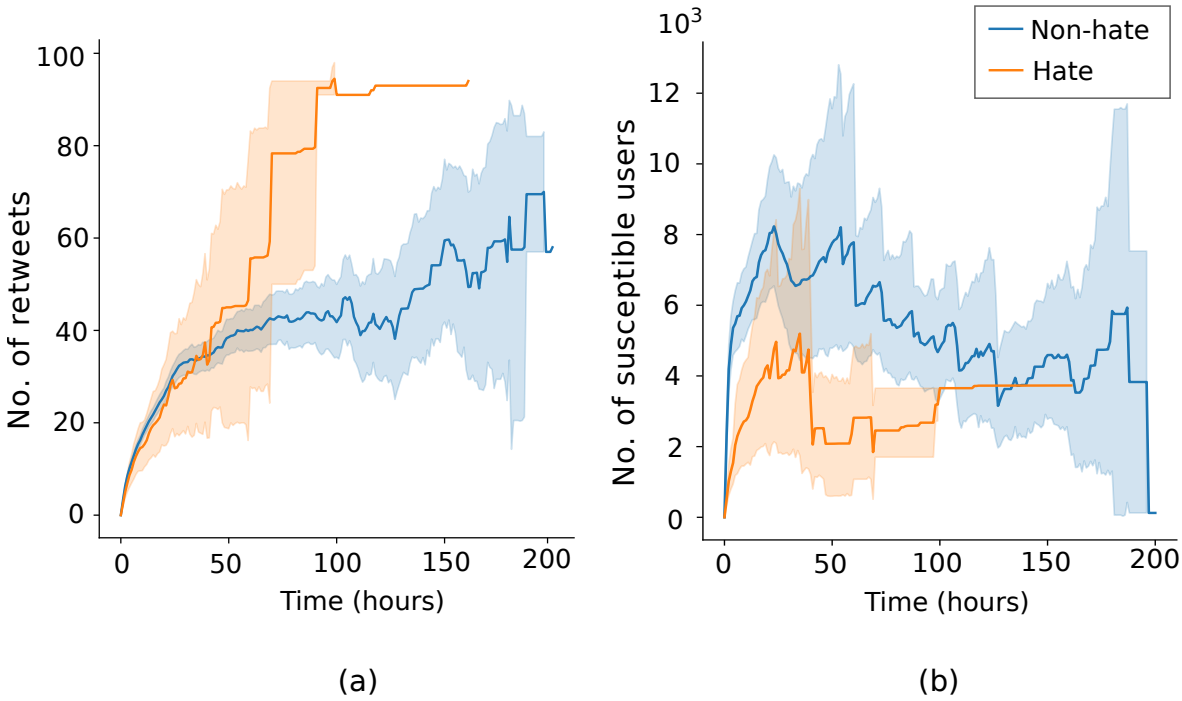

FIGURE 1.1: Plot (a) shows the growth of retweet cascades for hateful and non-hate tweets (solid lines and shaded regions signifying the average over the dataset and confidence of count, respectively). Analogously, plot (b) depicts the temporal change of susceptible users over time.

susceptible nodes at any time instance of the spread is defined by all such nodes which have been exposed to the information (followers of those who have posted/retweeted the tweet) up to that instant but did not participate in spreading (did not retweet/like/comment). While hateful tweets are retweeted in a significantly higher magnitude compared to non-hateful ones (see Figure 1.1(a)), they tend to create lesser number of susceptible users over time (see Figure 1.1(b)). This is directly linked to two major phenomena: primarily, one can relate this to the formation of hate echo-chambers – hateful contents are distributed among a well-connected set of users. Secondarily, as we define susceptibility in terms of follower relations, hateful contents, therefore, might have been diffusing among connections beyond the follow network – through paid promotion, etc. Also one can observe the differences in early growth for the two types of information; while hateful tweets acquire most of their retweets and susceptible nodes in a very short time and stall, later on, non-hateful ones tend to maintain the spread, though at a lower rate, for a longer time. This characteristic can again be linked to organized spreaders of hate who tend to disseminate hate as early as possible.

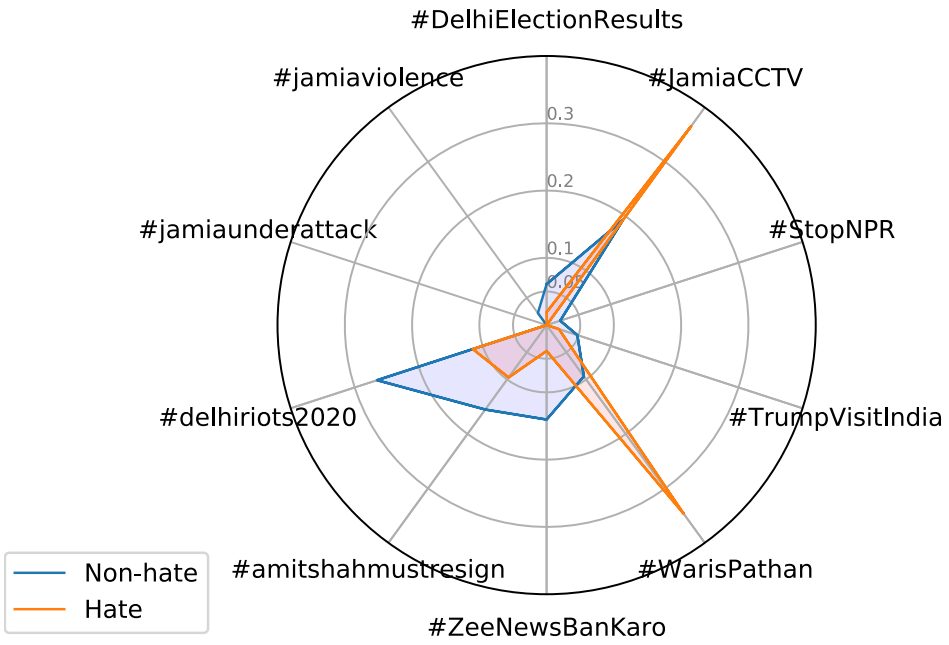

FIGURE 1.2: Distribution of hateful vs non-hate tweets (on a scale 0 to 1) for a selected number of hashtags.

Topic-dependence of Twitter hate. Hateful contents show strong topic-affinity: topics related to politics and social issues, for example, incur much more hateful content compared to sports or science. Hashtags in Twitter provide an overall mapping for tweets to topics of discussion. As shown in Figure 1.2, the degree of hateful content varies significantly for different hashtags. Even when different hashtags share a common theme (such as *of discussion jamiaunderattack, #jamiaviolence* and *#jamiaCCTV*), they may still incur a different degree of hate. Previous studies [5] tend to denote users as hate-preachers irrespective of the topic of discussion. However, as evident in Figure 1.3, the degree of hatefulness expressed by a user is dependent on the topic as well. For example, while some users resort to hate speech concerning COVID-19 and China, others focus on topics around the protests against the Citizenship Amendment Act in India.

Exogenous driving forces. With the increasing entanglement of virtual and real social processes, it is only natural that events happening outside the social media platforms tend to shape the platform's discourse. Though a small number of existing studies attempt to inquire into such inter-dependencies [6, 7], the findings are substantially motivating in problems related to modeling information diffusion and user engagement in Twitter and other platforms.

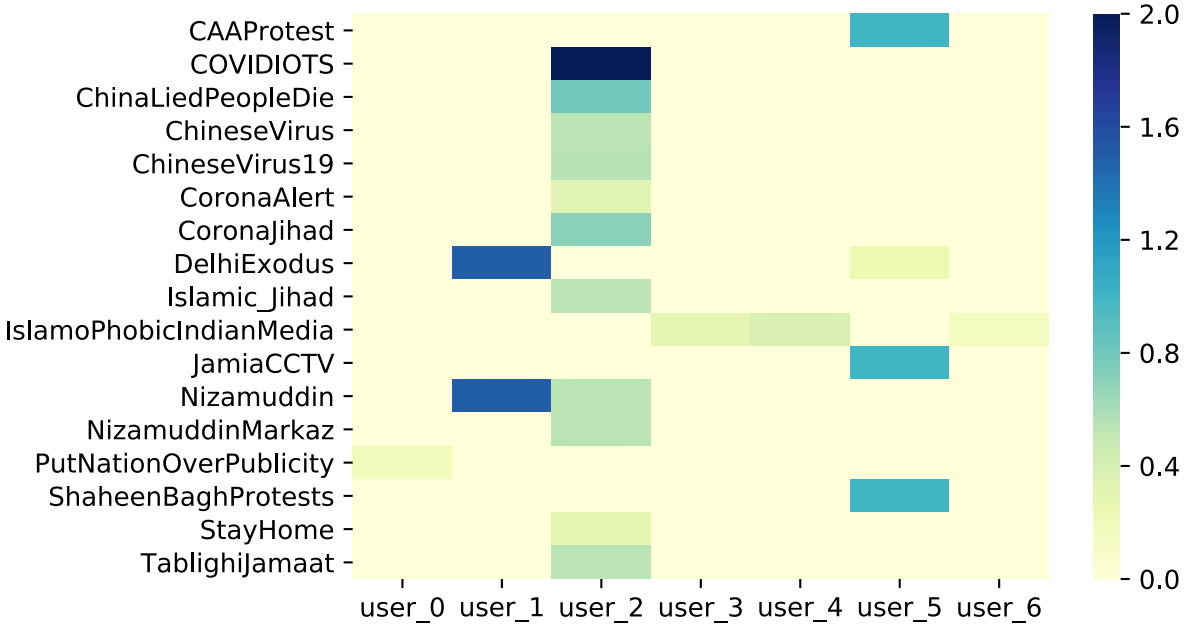

FIGURE 1.3: Distribution of hatefulness expressed by a selected set of users for different hashtags. The color of a cell corresponds to a user, and a hashtag signifies the ratio of hateful to non-hate tweets posted by that user using that specific hashtag.

In the case of hate speech, exogenous signals offer even more crucial attributes to look into, which is *global context*. For both detecting and predicting the spread of hate speech over short tweets, the knowledge of context is likely to play a decisive role (e.g., there has been $\sim$ 9-times rise in hate tweets aimed at Chinese people after the COVID-19 broke out[6]).

1.3 An Overview Of Research

Based on the findings of the existing literature and the analysis we presented above, here we attempt to model the dynamics of hate speech spread on Twitter. We separate the process of spread as the *hate generation* (asking for who will start a hate campaign) and *retweet diffusion of hate* (who will spread an already started hate campaign via retweeting). Our contributions can be summarized as follows:

(1) We formalize the dynamics of hate generation, and retweet spread on Twitter subsuming the activity history of each user and signals propagated by the localized structural

[6] https://shorturl.at/dJPS0

properties of the information network of Twitter induced by follower connections as well as global endogenous and exogenous signals (events happening inside and outside of Twitter) (See Section 4).

(2) We present a large dataset of tweets, retweets, user activity history, and the information network of Twitter covering versatile hashtags, which made to trend very recently. We manually annotate a significant subset of the data for hate speech. We also provide a corpus of contemporary news articles published online (see Section 3.1 for more details).

(3) We unsheathe rich set of features manifesting the signals mentioned above to design multiple prediction frameworks which forecast, given a user and a contemporary hashtag, whether the user will write a hateful post or not (Section 4.1). We provide an in-depth feature ablation and ensemble methods to analyze our proposed models' predictive capability, with the best performing one resulting in a macro F1 score of $\mathbf{0.65}$.

(4) We propose `RETINA` (**Ret**weeter **I**dentifier **N**etwork with Exogenous **A**ttention), a neural architecture to predict potential retweeters given a tweet (Section 4.2.2). `RETINA` encompasses an attention mechanism with dictates the prediction of retweeters based on a stream of contemporary news articles published online. Features representing hateful behavior encoded within the given tweet as well as the activity history of the users further help `RETINA` to achieve a macro F1 score of $\mathbf{0.85}$, significantly outperforming several state-of-the-art retweet prediction models.

CHAPTER 2

Related Work

Hate speech detection. In recent years, the research community has been keenly interested in better understanding, detection, and combating hate speech on online media. Starting with the basic feature-engineered logistic regression models[8, 9] to the latest ones employing neural architectures [10], variety of automatic online hate speech detection models have been proposed across languages [11]. To determine the hateful text, most of these models utilize a static-lexicon based approach and consider each post/comment in isolation. With lack of context (both in the form of individual's prior indulgence in the offense and the current world view), the models trained on previous trends, perform poorly on new datasets. While linguistic and contextual features are essential factors of a hateful message, the destructive power of hate speech lies in its ability to spread across the network. However, only recently have researchers started using network-level information for hate speech detection [12], [13].

While our work does not involve building a new hate speech detection model, yet hate detection underpins any work on hate diffusion in the first place. We have later compared the performance of the above mentioned hate detection classifiers on training the with our data and selected Davidson [9] as the best Hate Speech Detection classifier for our data. Inspired by existing research, we also incorporate hate lexicons as a feature for the diffusion model. The lexicon is curated from multiple sources and manually pruned to suit the Indian context [14]. Meanwhile, to overcome the problem of context, we utilize the whole timeline of a user, to determine her propensity towards hate speech.

Information diffusion and microscopic prediction. Predicting the spread of information on online platforms is crucial in understanding the network dynamics with applications in marketing campaigns, rumor spreading/stalling, route optimization, etc. The latest in the

family of diffusion being the CHASSIS [15] model. On the other end of the spectrum, the SIR model [16] effectively captures the presence of R (Recovered) nodes in the system, which are no longer active due to information fatigue[1]. Even though limited in scope, the SIR model serves as an essential baseline for all diffusion models.

Among other techniques, a host of studies employ social media data for both macroscopic (size and popularity) and microscopic (next user(s) in the information cascade) prediction. While highly popular, both DeepCas [17] and DeepHawkes [18] focus only on the size of the overall cascade. Similarly, Khosla et al. [19] utilized social cues to determine the popularity of an image on Flickr. While Independent Cascade (IC) based embedding models [20, 21] led the initial work in ML-based microscopic cascade prediction; they failed to capture the cascade's temporal history (either directly or indirectly). Meanwhile, Yang et al. [22] presented a neural diffusion model for microscopic prediction, which employs recurrent neural architecture to capture the history of the cascade. These models focus on predicting the next user in the cascade from a host of potential candidates. In this regard, TopoLSTM [23] considers only the previously seen nodes in any cascade as the next candidate without using timestamps as a feature. This approximation works well under limited availability of network information and the absence of cascade metadata. Meanwhile, FOREST [24] considers all the users in the global graph (irrespective of one-hop) as potential users, employing a time-window based approach. Work by Wang et al. [25] lies midway of TopoLSTM and FOREST, in that it does not consider any external global graph as input, but employs a temporal, two-level attention mechanism to predict the next node in the cascade. Zhou et al. [26] compiled a detailed outline of recent advances in cascade prediction.

Compared to the models discussed above for microscopic cascade prediction, which aim to answer who will be the next participant in the cascade, our work aims to determine whether a follower of a user will retweet (participate in the cascade) or not. This converts our use case into a binary classification problem, and adds negative sampling (in the form on inactive nodes), taking the proposed model closer to real-world scenario consisting of active and passive social media users.

[1] http://paginaspersonales.deusto.es/abaitua/konzeptu/fatiga.htm

The spread of hate and exploratory analysis by Mathew et al. [5] revealed exciting characteristics of the breadth and depth of hate vs. non-hate diffusion. However, their methodology separates the non-haters from haters and studies the diffusion of two cascades independently. Real-world interactions are more convoluted with the same communication thread containing hateful, counter-hateful, and non-hateful comments. Thus, independent diffusion studies, while adequate at the exploratory analysis of hate, cannot be directly extrapolated for predictive analysis of hate diffusion. The need is a model that captures the hate signals at the user and/or group level. By taking into account the user's timeline and his/her network traits, we aim to capture more holistic hate markers.

Exogenous influence. As early as 2012, Myers et al. [7] exposed that external stimuli drive one-third of the information diffusion on Twitter. Later, Hu et al. [27] proposed a model for predicting user engagement on Twitter that is factored by user engagement in 600 real-world events. From employing world news data for enhancing language models [28] to boosting the impact of online advertisement campaigns [29], exogenous influence has been successfully applied in a wide variety of tasks. Concerning social media discourse, both De et al. [30] in opinion mining and Dutta et al. [6] in chatter prediction corroborated the superiority of models that consider exogenous signals.

Since our data on Twitter was collected based on trending Indian hashtags, it becomes crucial to model exogenous signals, some of which may have triggered a trend in the first place. While a one-to-one mapping of news keywords to trending keywords is challenging to obtain, we collate the most recent (time-window) news w.r.t to a source tweet as our ground-truth.

CHAPTER 3

Datasets

3.1 Dataset Collection

Our collection of Twitter data spans from Monday 3rd February, 2020 to Tuesday 14th April, 2020. Using Twitter's official API[1], we tracked and crawled for trending hashtags each day within this duration. This resulted in a total of $20,625$ tweets from $13,965$ users. We also crawled the retweeters for each tweet along with the timestamps. Table 3.1 describes the hashtag-wise detailed statistics of the data. To build the information network, we collected the followers of each user up to a depth of 3, resulting in a total of $41,032,789$ unique users in our dataset. We also collect the activity history of the users, resulting in a total of $161,857,992$ tweets in our dataset.

We also, crawled the online news articles published within this span using the News-please crawler [31]. We managed to collect a total of $683,419$ news articles for this period. After filtering for language, title and date, we were left with $319,179$ processed items. There headlines were used as the source of exogenous signal.

#-tag acronyms for the #-tags mentioned in the Table 3.1:

JV: *jamiaviolence*, MOTR: *MigrantsOnTheRoad*, TTSV: *timetosackvadras*, JUA: *jamiaunderattack*, IBN: *IndiaBoycottsNPR*, ZNBK: *ZeeNewsBanKaro*, SCW: *SaluteCoronaWarriors*, IPIM: *IslamoPhobicIndianMedia*, DR2020: *delhiriots2020*, S4S: *Seva4Society*, PMCF: *PMCaresFunds*, C_19: *COVID_19*, HUA: *Hindus_Under_Attack*, WP: *WarisPathan*, LE:

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

TABLE 3.1: **Statistics of the data crawled from Twitter.** *Avg. RT*, *Users*, and *Users-all* signify average retweets, unique number of users tweeting and the unique number of users engaged in (tweet+retweet) the #-tag, respectively.

#-tags	JV	MOTR	TTSV	JUA	IBN	ZNBK	SCW
Tweets	950	872	280	263	570	919	104
Avg. RT	15.45	6.69	8.19	5.8	7.87	9.58	5.65
Users	743	641	138	215	333	751	53
Users-all	4026	2176	548	688	1227	1940	225
%-Hate	3.78%	8.20%	1.3%	6.06%	0.8%	7.01%	0.0%
#-tags	IPIM	DR2020	S4S	PMCF	C_19	HUA	WP
Tweets	1385	1453	1087	1172	971	382	989
Avg. RT	7.008	12.23	13.24	7.61	6.38	7.10	9.23
Users	842	1136	532	1076	807	292	807
Users-all	2934	6051	4058	2691	2593	1073	2924
%-Hate	13%	6.8%	1.53%	0.8%	1.96%	10.1%	12.07
#-tags	LE	JCCTV	TVI	PNOP	DE	DER	ASMR
Tweets	107	1045	339	555	542	843	959
Avg. RT	1.85	12.07	8.47	13.24	9.66	7.56	5.01
Users	102	815	284	365	414	731	765
Users-all	138	4091	1134	2146	1857	1807	1807
%-Hate	0.0%	5.66%	2.6%	5.71%	7.61%	3.20%	9.94%
#-tags	R4GK	DV	SNPR	1C4DH	NV	NM	90DSB
Tweets	949	1121	82	889	649	1124	226
Avg. RT	3.94	9.004	10.23	11.62	7.61	8.24	5.25
Users	492	948	64	770	546	843	188
Users-all	986	2702	440	3045	1577	3199	506
%-Hate	2.84%	7.37%	0.0%	0.99%	4.67%	7.85%	12.04%

lockdownextension, JCCTV: *JamiaCCTV*, TVI: *TrumpVisitIndia*, PNOP: *PutNationOverPublicity*, DE: *DelhiExodus*, DER: *DelhiElectionResults*, ASMR: *amitshahmustresign*, R4GK: *Restore4GinKashmir*, DV: *DelhiViolance*, SNPR: *StopNPR*, 1C4DH: *1Crore4DelhiHindu*, NV: *NirbhayaVerdict*, NM: *NizamuddinMarkaz*, 90DSB: *90daysofshaheenbagh*.

3.2 Data Preprocessing

We performed pre-processing on all the tweets before feeding them to our hate detection classifier as the performance may degrade due to the presence of hexadecimal characters and non-useful URLs. We used NLTK library and removed all the special characters, RT, cc, URLs, mentions and stopwords. This pre-processed clean text was then used everywhere.

3.3 Handling Suspended Users

While the data was collected, there were many accounts which were deactivated or suspended during that period. So, we could not gather history for those users. Hence, all those users and their tweets were not considered. It has been found that these users are often suspended when their tweets violated Twitter's content policy or related to sentiments or claims users decided to no longer promote. The removal of these users and their tweets has certainly decreased the hate content from our dataset.

CHAPTER 4

Hate Generation and Diffusion

An information network of Twitter can be defined as a directed graph $\mathcal{G} = \{\mathcal{U}, \mathcal{E}\}$, where every user corresponds to a unique node $u_i \in \mathcal{U}$, and there exists an ordered pair $(u_i, u_j) \in \mathcal{E}$ if and only if the user corresponding to u_j follows user u_i. (Table 4.1 summarizes important notations and denotations.) Typically, the visible information network of Twitter does not associate the follow relation with any further attributes, therefore any two edges in $\mathcal{E}$ are indistinguishable from each other. We associate unit weight to every $e \in \mathcal{E}$.

Every user in the network acts as an agent of content generation (tweeting) and diffusion (retweeting). For every user u_i at time t_0, we associate an activity history $\mathcal{H}_{i,t_0} = \{\tau(t) | t \leq t_0\}$, where $\tau(t)$ signifies a tweet posted (or retweeted) by u_i at time t.

The information received by user u_i has three different sources: (a) *Peer signals* ($\mathcal{S}_i^P$): The information network $\mathcal{G}$ governs the flow of information from node to node such that any tweet posted by u_i is visible to every user u_j if $(u_i, u_j) \in \mathcal{E}$; (b) *Non-peer endogenous signals* ($\mathcal{S}^{\text{en}}$): Trending hashtags, promoted contents, etc. that show up on the user's feed even in the absence of peer connection; (c) *Exogenous signals* ($\mathcal{S}^{\text{ex}}$): Apart from the Twitter feed, every

TABLE 4.1: Important notations and denotations.

Notation	Denotation
$\mathcal{G}$	The information network
$\mathcal{H}_{i,t}$	Activity history of user u_i up to time t
$\mathcal{S}^{\text{ex}}$	Exogenous influence
$\mathcal{S}^{\text{en}}$	Endogenous influence
$\mathcal{S}_i^P$	Peer influence on u_i
$\mathcal{T}$	Topic (hashtag)
P^{u_i}, $P_j^{u_i}$	Probability of u_i retweeting (static vs. j^{th} interval)
$\mathbf{X}^T$, $\mathbf{X}^N$	Feature tensors for tweet and news
$\mathbf{X}^{T,N}$	Output from exogenous attention

user interacts with the external world-events directly (as a participant) or indirectly (via news, blogs, etc.).

Hate generation. The problem of modeling hate generation can be formulated as assigning a probability with each user that signifies their likelihood to post a hateful tweet. With our hypothesis of hateful behavior being a topic-dependent phenomenon, we formalize the modeling problem as learning the parametric function, $f_1 : \mathbb{R}^d \rightarrow (0, 1)$ such that,

$$P(u_i|\mathcal{T}) = f_1(\mathcal{S}^{\text{en}}, \mathcal{S}^{\text{ex}}, \mathcal{H}_{i,t}, \mathcal{T}|\theta_1) \tag{4.1}$$

where $\mathcal{T}$ is a given topic, t is the instance up to which we obtain the observable history of u_i, d is the dimensionality of the input feature space, and θ_1 is the set of learnable parameters. Though ideally $P(u_i|\mathcal{T})$ should be dependent on $\mathcal{S}_i^P$ as well, the complete follower network for Twitter remains mostly unavailable due to account settings, privacy constraints, inefficient crawling, etc.

Hate diffusion. As already stated, we characterize diffusion as the dynamic process of retweeting in our context. Given a tweet $\tau(t_0)$ posted by some user u_i, we formulate the problem as predicting the potential retweeters within the interval $[t_0, t_0 + \Delta t]$. Assuming the probability density of a user u_j retweeting τ at time t to be $p(t)$, then retweet prediction problem translates to learning the parametric function $f_2 : \mathbb{R}^d \rightarrow (0, 1)$ such that,

$$\int_{t_0}^{t_0+\Delta t} p(t)dt = f_2(\mathcal{S}_j^P, \mathcal{S}_j^{\text{en}}, \mathcal{S}_j^{\text{ex}}, \mathcal{H}_{j,t}, \tau|\theta_2) \tag{4.2}$$

Eq. 4.2 is the general form of a parametric equation describing retweet prediction. In our setting, the signal components $\mathcal{S}_j^P$, $\mathcal{H}_{j,t}$, and the features representing the tweet τ incorporates the knowledge of hatefulness. Henceforth, we call τ the *root tweet* and u_i the *root user*. It is to be noted that, the features representing the peer, non-peer endogenous, and exogenous signals in Eq. 4.1 and 4.2 may differ due to the difference in problem setting.

Beyond organic diffusion. The task of identifying potential retweeters of a post on Twitter is not straightforward. In retrospect, the event of a user retweeting a tweet implies that the user must have been an audience of the tweet at some point of time (similar to 'susceptible' nodes

of contagion spread in the SIR/SIS models [16],[32]). For any user, if at least one of his/her followees engages with the retweet cascade, then the subject user becomes susceptible. That is, in an *organic* diffusion, between any two users u_i, u_j there exists a finite path $\langle u_i, u_{i+1} \ldots, u_j \rangle$ in $\mathcal{G}$ such that each user (except u_i) in this path is a retweeter of the tweet by u_i. However, due to account privacy etc., one or more nodes within this path may not be visible. Moreover, contents promoted by Twitter, trending topics, content searched by users independently may diffuse alongside their organic diffusion path. Searching for such retweeters is impossible without explicit knowledge of these phenomena. Hence, we primarily restrict our retweet prediction to the organic diffusion, though we experiment with retweeters not in the visibly organic diffusion cascade to see how our models handle such cases.

4.1 Modelling Hate Generation

To realize Eq. 4.1, we signify topics as individual hashtags. We rely purely on manually engineered features for this task so that rigorous ablation study and analysis produce explainable knowledge regarding this novel problem. The extracted features instantiate different input components of f_1 in Eq. 4.1. We formulate this task in a static manner, i.e., assuming that we are predicting at an instance t_0, we want to predict the probability of the user posting a hateful tweet within $[t_0, \infty]$. While training and evaluating, we set t_0 to be right before the actual tweeting time of the user.

4.1.1 User history-based features

The activity history of user u_i, signified by $\mathcal{H}_{i,t}$ is substantiated by the following features:

• We use unigram and bigram features weighted by tf-idf values from 10 most recent tweets posted by u_i to capture its recent topical interest. To reduce the dimensionality of the feature space, we keep the top 300 features sorted by their idf values.

• To capture the history of hate generation by u_i, we compute two different features her most recent 10 tweets: (i) ratio of hateful vs. non-hate tweets and (ii) a hate lexicon vector

$HL = \{h_i | h_i \in \mathbb{I}^+ \text{ and } i = 1, \ldots, |\mathbf{H}|\}$, where $\mathbf{H}$ is a dictionary of hate words, and h_i is the frequency of the i^{th} lexicon from $\mathbf{H}$ among the tweet history.

• Users who receive more attention from the fellow users for hate propagation are more likely to generate hate. Therefore, we take the ratio of retweets of previous hateful tweets to non-hateful ones by u_i. We also take the ratio of total number of retweets on hateful and non-hateful tweets of u_i .

• Follower count and date of account creation of u_i.

• Number of topics (hashtags) u_i has tweeted on up to t.

4.1.2 Topic (hashtag)-oriented feature

We compute Doc2Vec [33] representations of the tweets, along with the hashtags present in them as individual tokens. We then compute the average cosine similarity between the user's recent tweets and the word vector representation of the hashtag, this serves as the topical relatedness of the user towards the given hashtag.

4.1.3 Non-peer endogenous features

To incorporate the information of trending topics over Twitter, we supply the model with a binary vector representing the top 50 trending hashtags for the day the tweet is posted.

4.1.4 Exogenous feature

We compute the average tf-idf vector for the 60 most recent news headlines from our corpus posted before the time of the tweet. Again we select the top 300 features.

Using the above features, we implement six different classification models(and their variants). Details of the models are provided in Section 5.0.2.

4.2 Retweet Prediction

While realizing Eq. 4.2 for retweeter prediction, we formulate the task in two different settings: the *static retweeter prediction* task, where t_0 is fixed, and Δt is ∞ (i.e., all the retweeters irrespective of their retweet time) and the *dynamic retweeter prediction* task where we predict on successive time intervals.

For these tasks, we rely on features both designed manually as well as extracted using unsupervised/self-supervised manner.

4.2.1 Feature selection

For the task of retweet prediction, we extract features representing the root tweet itself, as well as the signals of Eq. 4.2 corresponding to each user u_i (for which we predict the possibility of retweeting). Henceforth, we indicate the root user by u_0.

Here, we incorporate $\mathcal{S}_i^P$ using two different features: shortest path length from u_0 to u_i in $\mathcal{G}$, and number of times u_i has retweeted tweets by u_0. All the features representing $\mathcal{H}_{i,t}$ and $\mathcal{S}^{en}$ remain same as described in Section 4.1.

We incorporate two sets of features representing the root tweet τ: the hate lexicon vector similar to Section 4.1.1 and top 300 unigram and bi-gram features weighted by tf-idf values.

For the retweet prediction task, we incorporate the exogenous signal in two different methods. To implement the attention mechanism of `RETINA`, we use a Doc2Vec representations of the news articles as well as the root tweet. For rest of the models, we use the same feature set as Section 4.1.4.

4.2.2 Design of `RETINA`

Guided by Eq. 4.2, `RETINA` exploits the features described in Section 4.2.1 for both static and dynamic prediction of retweeters.

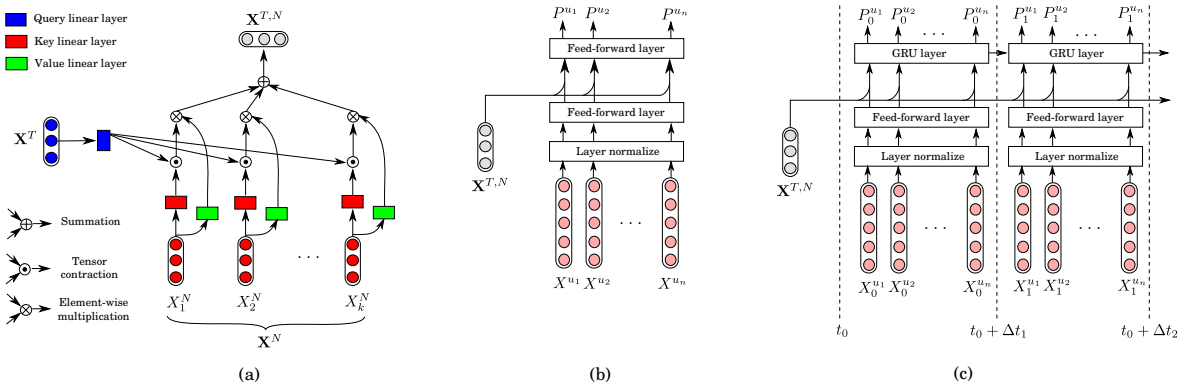

FIGURE 4.1: **Design of different components of RETINA** – (a) *Exogenous attention*: Key and Value linear layers (blue) are applied on each element of the news feature sequence $\mathbf{X}^N$, while the Query linear layer (red) is applied on the tweet feature $\mathbf{X}^T$. The attention weights computed for each news feature vector by contracting the query and key tensors along feature axis (dot product) are then applied to the value tensors and summed over the sequence axis to produce the 'attended' output, $\mathbf{X}^{T,N}$. (b) *Static prediction of retweeters*: To predict whether u_j will retweet, the input feature X^{u_j} is normalized and passed through a feed-forward layer, concatenated with $\mathbf{X}^{T,N}$, and another feed-forward layer is applied to predict the retweeting probability P^{u_j}. (c) *Dynamic retweet prediction*: In this case, RETINA predicts the user retweet probability for consecutive time intervals, and instead of the last feed-forward layer used in the static prediction, we use a GRU layer.

Exogenous attention. To incorporate external information as an assisting signal to model diffusion, we use a variation of *scaled dot product attention* [34] in RETINA (see Figure 4.1).

Given the feature representation of the tweet $\mathbf{X}^T$ and news feature sequence $\mathbf{X}^N = \{X_1^N, X_2^N, \ldots, X_k^N\}$, we compute three tensors $\mathbf{Q}^T$, $\mathbf{K}^N$, and $\mathbf{V}^N$, respectively as follows:

$$\begin{aligned} \mathbf{Q}^T &= \mathbf{X}^T \odot |_{(-1,0)} \mathbf{W}^Q \\ \mathbf{K}^N &= \mathbf{X}^N \odot |_{(-1,0)} \mathbf{W}^K \\ \mathbf{V}^N &= \mathbf{X}^N \odot |_{(-1,0)} \mathbf{W}^V \end{aligned} \tag{4.3}$$

where $\mathbf{W}^Q$, $\mathbf{W}^K$, and $\mathbf{W}^V$ are learnable parameter kernels (we denote them to belong to *query*, *key* and *value* dense layers, respectively in Figure 4.1). The operation $(\cdot) \odot |_{(-1,0)} (\cdot)$ signifies *Tensor contraction* according to *Einstein summation convention* along the specified axis. In Eq. 4.3, $(-1, 0)$ signifies last and first axis of the first and second tensor, respectively.

Therefore, $X \odot |_{(-1,0)} Y = \sum_i X[\dots, i] Y[i, \dots]$. Each of $\mathbf{W}^Q$, $\mathbf{W}^K$, and $\mathbf{W}^V$ is a two-dimensional tensor with $hdim$ columns (last axis).

Next, we compute the attention weight tensor $\mathbf{A}$ between the tweet and news sequence as

$$\mathbf{A} = Softmax(\mathbf{Q}^T \odot |_{(-1,-1)} \mathbf{K}^N) \tag{4.4}$$

where $Softmax(X[\dots, i, j]) = \frac{e^{X[\dots,i,j]}}{\sum_j e^{X[\dots,i,j]}}$. Further, to avoid saturation of the softmax activation, we scale each element of $\mathbf{A}$ by $hdim^{-0.5}$ [34].

The attention weight is then used to produce the final encoder feature representation $\mathbf{X}^{T,N}$ by computing the weighted average of $\mathbf{V}^N$ as follows:

$$\mathbf{X}^{T,N} = \sum_i \mathbf{V}^N[\dots, i, \dots] \mathbf{A}[\dots, i] \tag{4.5}$$

RETINA is expected to aggregate the exogenous signal exposed by the sequence of news inputs according to the feature representation of the tweet into $\mathbf{X}^{T,N}$, using the operations mentioned in Eqs. 4.3-4.5 via tuning the parameter kernels.

Final prediction. With $\mathcal{S}^{\text{ex}}$ being represented by the output of the attention framework, we incorporate the features discussed in Section 4.2.1 in RETINA to subsume rest of the signals (see Eq. 4.2). For the two separate modes of retweeter prediction (i.e., static and dynamic), we implement two different variations of RETINA.

For the static prediction of retweeters, RETINA predicts the probability of each of the users $u_1, u_2, \dots, u_n$ to retweet the given tweet with no temporal ordering (see Figure 4.1 (b)). The feature vector X^{u_i} corresponding to user u_i is first normalized and mapped to an intermediate representation using a feed-forward layer. It is then concatenated with the output of the exogenous attention component, $\mathbf{X}^{T,N}$, and finally another feed-forward layer with sigmoid nonlinearity is applied to compute the probability P^{u_i}.

As opposed to the static case, in the dynamic setting RETINA predicts the probability of every user u_i to retweet within a time interval $t_0 + \Delta t_i, t_0 + \Delta t_{i+1}$, with t_0 being the time of the tweet published and $\Delta t_0 = 0$. To capture the temporal dependency between predictions

in successive intervals, we replace the last feed-forward layer with a Gated Recurrent Unit (GRU)[1], as shown in Figure 4.1 (c).

Cost/loss function. In both the settings, the task translates to a binary classification problem of deciding whether a given user will retweet or not. Therefore, we use standard *binary cross-entropy loss* L to train RETINA:

$$L = -w \cdot t \log(p) - (1 - t) \log(1 - p) \tag{4.6}$$

where t is the ground-truth, p is predicted probability (P^{u_i} in static and $P_j^{u_i}$ in dynamic settings), and w is a the weight given to the positive samples to deal with class imbalance.

[1]We experimented with other recurrent architectures as well; performance degraded with simple RNN and no gain with LSTM.

CHAPTER 5

Experiments, Results and Analysis

5.0.1 Detecting hateful tweets

We employ three professional annotators who have experience in analyzing online hate speech to annotate the tweets manually. We annotated a total of $17,877$ tweets with an inter-annotator agreement of 0.58 Krippendorf's α[1]. We select the final tags based on majority voting.

Based on this gold-standard annotated data, we train three different hate speech classifiers based on the designs given by Davidson et al. [9], Waseem and Hovy [8], and Pinkesh et al. [10]. With an AUC score 0.85 and macro-F1 0.59, the Davidson et al. model emerges as the best performing one. We use this model to annotate rest of the tweets in our dataset (% of hateful tweets for each hashtag is reported in Table 3.1). We use the machine-annotated tags for the features and training labels in our proposed models only, while the hate generation models are tested solely on gold-standard data.

Along with the manual annotation and trained hate detection model, we use a dictionary of hate lexicons proposed in [14]. It contain a total of 209 words/phrases signaling a possible existence of hatefulness in a tweet.

5.0.2 Hate generation

To experiment on our hate generation prediction task, we use a total of $19,032$ tweets coming from $12,492$ users to construct the ground-truth. With an $80:20$ train-test split, there are

[1]The low value of inter-annotator's agreement is at par with most hate speech annotation till date, pointing out the hardness of the task even for human subjects. This further strengthens the need for contextual knowledge as well as exploiting beyond-the-text dynamics.

TABLE 5.1: **List of model parameters used for predicting hate generation.** *SVM-r* and *SVM-l* refer to Support Vector Machine with rbf and linear kernels, respectively. *LogReg*: Logistic Regression, *Dec-Tree*: Decision Tree.

Classifier	Parameters
LogReg	Random state=0
AdaBoost	Random State=1
SVM-r	Class Weight = 'Balanced'
SVM-l	Penalty= l2, Class Weight = 'Balanced'
Dec-Tree	Class Weight = 'Balanced', Max Depth = 5
XGBoost	eta=0.4, eval metric= 'logloss', learning rate=0.0001, objective= 'binary:logistic', reg alpha = 0.9

611 hateful tweets among $15,225$ in the training data, whereas 129 out of $3,807$ in the testing data.

To deal with the severe class imbalance of the dataset, we use both upsampling of positive samples and downsampling of negative samples.

With all the features discussed in Section 4.1, the full size of the feature vector is $3,645$. We experimented with all our proposed models with this full set of features and dimensionality reduction techniques applied to it. We use Principal Component Analysis (PCA) with the number of components set to 50. Also, we conduct experiments selecting K-best features ($K = 50$) using mutual information.

We implement a total of six different classifiers using Support Vector Machine (with linear and RBF kernel), Logistic Regression, Decision Tree, AdaBoost, and XGBoost [35]. Parameter settings for each of these are reported in Table 5.1. All of the models, PCA, and feature section are implemented using scikit-learn[2].

5.0.3 Retweeter prediction

The activity of retweeting, too, shows a skewed pattern similar to hate speech generation. While the maximum number retweets for a single tweet is 133 in our dataset, the average remains to be 11.97. We use only those tweets with more than one retweet for all of the

[2] https://scikit-learn.org/stable/

proposed models. With an $80:20$ train-test split, this results in a total of $3,057$ and 765 samples for training and testing.

For all the Doc2Vec generated feature vectors related to tweets and news headlines, we set the dimensionality to 50 and 500, respectively. For RETINA, we set the parameter $hdim$ and all the intermediate hidden sizes for the rest of the feed-forward (except the last one generating logits) and recurrent layers to 64 (see Section 4.2.2).

Hyperparameter tuning of RETINA. For both the settings (i.e, static and dynamic prediction of retweeters), we used mini-batch training of RETINA, with both Adam and SGD optimizers. We varied the batch size within 16, 32 and 64, with the best results for a batch size of 16 for the static mode and 32 for the dynamic mode. We also varied the learning rates within a range 10^{-4} to 10^{-1}, and chose the best one with learning rate 10^{-2} using the SGD optimizer[3] for the dynamic mode. The static counterpart produced the best results with Adam optimizer[4] [36] using default parameters.

To deal with the class imbalance, we set the parameter w in Eq. 4.6 as $w = \lambda(\log C - \log C^{+})$, where C and C^{+} are the counts for total and positive samples, respectively in the training dataset, and λ is a balancing constant which we vary from 1 to 2.5 with 0.5 steps. We found the best configurations with $\lambda = 2.0$ and $\lambda = 2.5$ for the static and dynamic modes respectively.

5.1 Baselines and ablation variants

In the absence of external baselines for predicting hate generation probability due to the problem's novelty, we explicitly rely on ablation analyses of the models proposed for this task. For retweet dynamics prediction, we implement 5 external baselines and two ablation variants of RETINA. Since information diffusion is a vast subject, we approach it from two perspectives – one is the set of rudimentary baselines (SIR, General Threshold), and the other is the set of recently proposed neural models.

[3] https://www.tensorflow.org/api_docs/python/tf/keras/optimizers/SGD
[4] https://www.tensorflow.org/api_docs/python/tf/keras/optimizers/Adam

TABLE 5.2: **Evaluation of different classifiers for the prediction of hate generation.** *Proc.* signifies different feature selection and label sampling methods, where *DS*: downsampling of dominant class, *US*: upsampling of dominated class, *PCA*: feature dimensionality reduction using PCA, *top-K*: selecting top-K features with $K = 50$.

Model	Proc.	m-F1	ACC	AUC	Model	Proc.	m-F1	ACC	AUC	Model	Proc	m-F1	ACC	AUC
	None	0.52	0.94	0.52		None	0.55	0.88	0.61		None	0.50	0.96	0.503
	DS	0.63	0.73	0.63		DS	0.62	0.70	0.64		DS	0.64	0.79	0.629
SVM	US+DS	0.44	0.64	0.63	SVM	US+DS	0.46	0.69	0.66	LogReg	US+DS	0.47	0.72	0.63
linear	PCA	0.55	0.90	0.59	rbf	PCA	0.48	0.71	**0.68**		PCA	0.49	0.97	0.50
	top-K	0.53	0.84	0.63		top-K	0.50	0.79	0.62		top-K	0.49	0.97	0.50
	PCA+DS	0.57	0.71	0.57		PCA+DS	0.63	0.68	0.66		PCA+DS	0.60	0.78	0.59
	None	0.51	0.79	0.64		None	0.49	0.97	0.49		None	0.53	0.97	0.52
	DS	**0.65**	0.74	0.66		DS	0.62	0.77	0.61		DS	0.57	0.76	0.566
Dec-	US+DS	0.45	0.67	0.61	Ada-	US+DS	0.44	0.63	**0.68**	XGB	US+DS	0.44	0.66	0.62
Tree	PCA	0.46	0.68	0.65	Boost	PCA	0.50	0.97	0.50		PCA	0.51	0.96	0.51
	top-K	0.53	0.84	0.63		top-K	0.49	0.97	0.50		top-K	0.49	0.97	0.50
	PCA+DS	0.60	0.66	0.63		PCA+DS	0.61	0.78	0.59		PCA+DS	0.56	0.73	0.55

5.1.1 Rudimentary baselines

SIR: The Susceptible-Infectious-Recovered (Removed) [16] is one of the earliest predictive models for contagion spread. It is an epidemiological model that computes the number of people infected with a contagious illness in a closed population over time. The name of this class of models derives from the fact that they involve coupled equations relating the number of susceptible people S(t), number of people infected I(t), and number of people who have recovered R(t).

In terms of information diffusion, SIR can be modelled as, a node is in the susceptible state 'S', when it has not been influenced by the information but may get infected by nearby infected nodes. A node becomes infected with state 'I', when it has been influenced by the information and when it can spread the information to other nodes. And, a node is in the recovered 'R' state, when a node can not be influenced by the information again. A node gets infected by its friends and nearby nodes by information in a similar way as in the epidemics.

Threshold Model: This model assumes that each node has threshold inertia chosen uniformly at random from the interval $[0, 1]$. A node becomes active if the weighted sum of its active neighbors exceeds this threshold. [37]

In simple terms, in the LT model, each edge has a weight, each vertex has a threshold chosen uniformly at random, and a vertex becomes activated if the weighted sum of its active neighbors exceeds its threshold.[38]

5.1.2 Neural network based baselines

To overcome the feature engineering step involving combinations of topical, contextual, network, and user-level features, neural methods for information diffusion have gained popularity. While these methods are all focused on determining only the next set of users, they are still important to measure the diffusion performance of RETINA.

TopoLSTM [23]: It is one of the initial works to consider recurrent models in generating the next user prediction probabilities. The model converts the cascades into dynamic DAGs (capturing the temporal signals via node ordering). The sender-receiver based RNN model captures a combination of active node's static score (based on the history of the cascade), and a dynamic score (capturing future propagation tendencies).

It takes dynamic DAGs as inputs and generates topological aware embedding for each node in the DAGS as outputs.The cascades are based in the order of the nodes in each cascade sequence and thus can be written as $s = (v_1, 1)(v_2, 1), (v_{t-1}, T)$. And with these cascade sequences the diffusion model is able to predict a node to activate at time t , given a test sequence $s' = (v'_1, 1)(v'_2, 1), (v'_{t-1}, T)$. Also the senders embedding h_t can be learned recursively from the earlier $h'_i s$, as they all store the encoded information about the diffusion topology before time t.

They have extended the standard LSTM to TOPO-LSTM for modelling the diffusion topologies., which are DAGs. It has 3 types of inputs (1) $v'_t s$ feature vector x_t . (2)possible activation attempts from $v'_t s$ precedent set $P_{v,.t}$, (3) The other activated nodes $Q_{1:t-1}/P_{v,t}$. The major difference it has from normal LSTM is it has different types of inputs and multiple inputs in each type.

FOREST [24]: It aims to be a unified model, performing the microscopic and the macroscopic cascade predictions combining reinforcement learning (for macroscopic) with the recurrent model (for microscopic). By considering the complete global graph, it performs graph sampling to obtain the structural context of a node as an aggregate of the structural context of its one or two hops neighbors. In addition, it factors the temporal information via the last m seen nodes in the cascade. The sender-receiver based RNN model is enhanced by the action-state model of RL, to give superior performance for both the tasks. It is worth noticing that largest graph the model was tested on consisted of $\approx 23k$ nodes, while even after sampling our smallest graph has $\approx 150k$ nodes. This six-fold increase in data, coupled with limited computing power, drastically reduced the performance of FOREST, as it tries to predict next user probabilities against all $150k$ nodes.

HIDAN [25]: It does not explicitly consider a global graph as input. Any information loss due to the absence of a global graph is substituted by temporal information utilized in the form of ordered time difference of node infection. The model uniquely captures the non-sequential nature of information diffusion, by employing a two-tier attention mechanism. The first attention system, builds the structural context of a user from the historical inter-user interactions. This context is aggregated into user's own context. Meanwhile, at cascade-level the temporal information is used, along with the updated user context to predict the next node. Since HIDAN does not employ a global graph, like TopoLSTM, it too uses the set of all seen nodes in the cascade as candidate nodes for prediction.

5.1.3 Ablation models

In the absence of external baselines for predicting hate generation probability due to the problem's novelty, We exercise extensive feature ablation to examine the relative importance of different feature sets. Among the six different algorithms we implement for this task, along with different sampling and feature reduction methods, we choose the best performing model for this ablation study. Following Eq. 4.1, we remove the feature sets representing $\mathcal{H}_{i,t}$, $\mathcal{S}^{\text{ex}}$, $\mathcal{S}^{\text{en}}$, and $\mathcal{T}$ (see Section 4.1 for corresponding features) in each trial and evaluate the performance.

TABLE 5.3: **Feature ablation for Decision Tree with downsampling for predicting hate generation.** At each trial, we remove features representing signals – $\mathcal{H}_{i,t}$ (*All* $\setminus$ *History*), $\mathcal{S}^{\text{ex}}$ (*All* $\setminus$ *Endogen*), $\mathcal{S}^{\text{en}}$ (*All* $\setminus$ *Exogen*), and $\mathcal{T}$ (*All* $\setminus$ *Topic*). See Eq. 4.1 and Section 4.1 for details of the signals and features, respectively.

Features	m-F1	ACC	AUC
All	0.65	0.74	0.66
All $\setminus$ History	0.56	0.59	0.64
All $\setminus$ Endogen	0.61	0.68	0.64
All $\setminus$ Exogen	0.56	0.58	0.66
All $\setminus$ Topic	0.65	0.74	0.66

To investigate the effectiveness of the exogenous attention mechanism for predicting potential retweeters, we remove this component and experiment on static as well as the dynamic setting of RETINA.

5.2 Evaluation

Evaluation of classification models on highly imbalanced data needs careful precautions to avoid classification bias. We use multiple evaluation metrics for both the tasks: macro averaged F1 score (macro-F1), area under the receiver operating characteristics (AUC), and binary accuracy (ACC). As the neural baselines tackle the problem of retweet prediction as a ranking task, we improvise the evaluation of RETINA to make it comparable with these baselines. We rank the predicted probability scores (P^{u_i} and $P_j^{u_i}$ in static and dynamic settings, respectively) and compute mean average precision at top-k positions (MAP@k) and binary hits at top-k positions (HITS@k).

5.2.1 Performance in predicting hate generation

Table 5.2 presents the performances of all the models we implement to predict the probability of a given user posting a hateful tweet using a given hashtag. It is evident from the results that, all six models suffer from the sharp bias in data; without any class-specific sampling, they tend to lean towards the dominant class (non-hate in this case) and result in a low macro-F1

and AUC compared to very high binary accuracy. SVM with rbf-kernel outperforms the rest when no upsampling or downsampling is done, with a macro-F1 of 0.55 (AUC 0.61).

Effects of sampling. Downsampling the dominant classes result in a substantial leap in the performance of all the models. The effect is almost uniform over all the classifiers except XGBoost. In terms of macro-F1, Decision Tree sets the best performance altogether for this task as $\mathbf{0.65}$. However, the rest of the models lie in a very close range of 0.62-0.64 macro-F1.

While the downsampling performance gains are explicitly evident, the effects of upsampling the dominated class are less intuitive. For all the models, upsampling deteriorates macro-F1 by a large extent, with values in the range 0.44-0.47. However, the AUC scores improve by a significant margin for all the models with upsampling except Decision Tree. AdaBoost achieves the highest AUC of $\mathbf{0.68}$ with upsampling.

Dimensionality reduction of feature space. Our experiments with PCA and K-best feature selection by mutual information show a heterogeneous effect on different models. While the only SVM with linear kernel shows some improvement with PCA over the original feature set, the rest of the models observe considerable degradation of macro-F1. However, SVM with rbf kernel achieves the best AUC of 0.68 with PCA. With top-K best features, the overall gain in performance is not much significant except Decision Tree.

We also experiment with combinations of different sampling and feature reduction methods, but none of them achieve a significant gain in performance.

Ablation analysis. We choose Decision Tree with down-sampling of dominant class as our best performing model (in terms of macro-F1 score) and perform ablation analysis. Table 5.3 presents the performance of the model with each feature group removed in isolation, along with the full model. Evidently, for predicting hate generation, **features representing exogenous signals and user activity history are most important**. Removal of the feature vector signifying trending hashtags, which represent endogenous signal in our case, also worsens the performance to a significant degree.

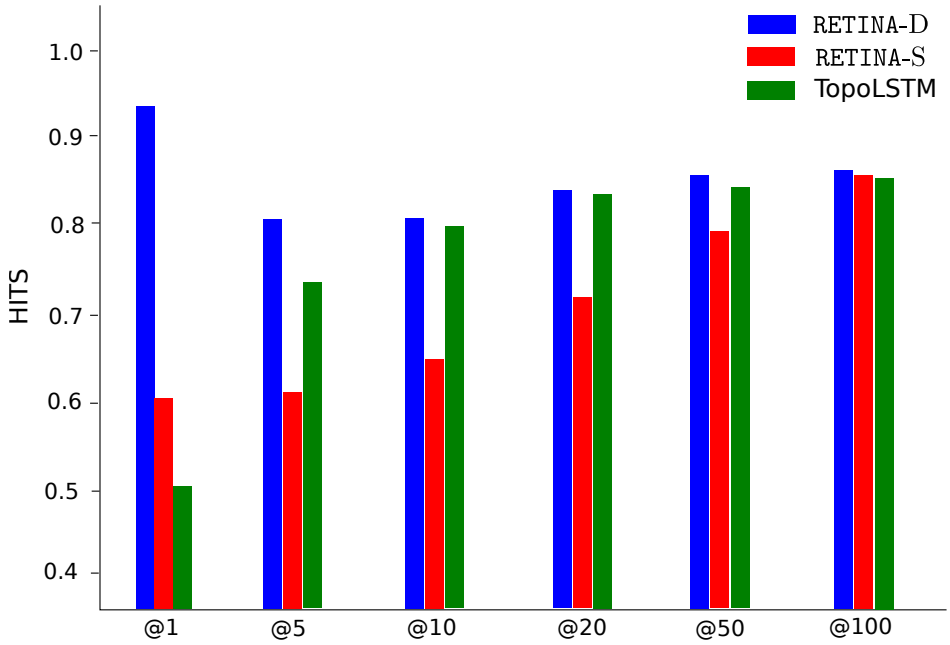

FIGURE 5.1: HITS@k of RETINA-D, RETINA-S, and TopoLSTM for retweeter prediction with $k = 1, 5, 10, 20, 50$, and 100.

TABLE 5.4: **Performance comparison for RETINA with baseline models for retweeter prediction.** RETINA-S and RETINA-D correspond to static and dynamic prediction settings, respectively. † symbolizes RETINA (static and dynamic) without exogenous attention. *Gen.Thresh.* corresponds to the General Threshold model for diffusion prediction.

Model	m-F1	ACC	AUC	MAP@20	HITS@20
LReg-S	0.61	0.92	0.682		
RETINA-S	0.65	0.95	0.75	0.54	0.72
RETINA-S†	0.61	0.92	0.77	0.53	0.71
LReg-D	0.82	.93	.79		
RETINA-D	**0.89**	0.99	**0.86**	**0.78**	**0.88**
RETINA-D†	0.87	0.99	0.798	0.69	0.80
FOREST	-	-	-	0.51	0.64
HIDAN	-	-	-	0.05	0.05
TopoLSTM	-	-	-	0.60	0.83
SIR	0.04	-	-	-	-
Gen.Thresh.	0.04	-	-	-	-

5.2.2 Performance in retweeter prediction

Table 5.4 summarizes the performances of the competing models for the retweet prediction task. Here again, binary accuracy presents a very skewed picture of the performance due to class imbalance. While RETINA in dynamic setting outperforms rest of the models by a

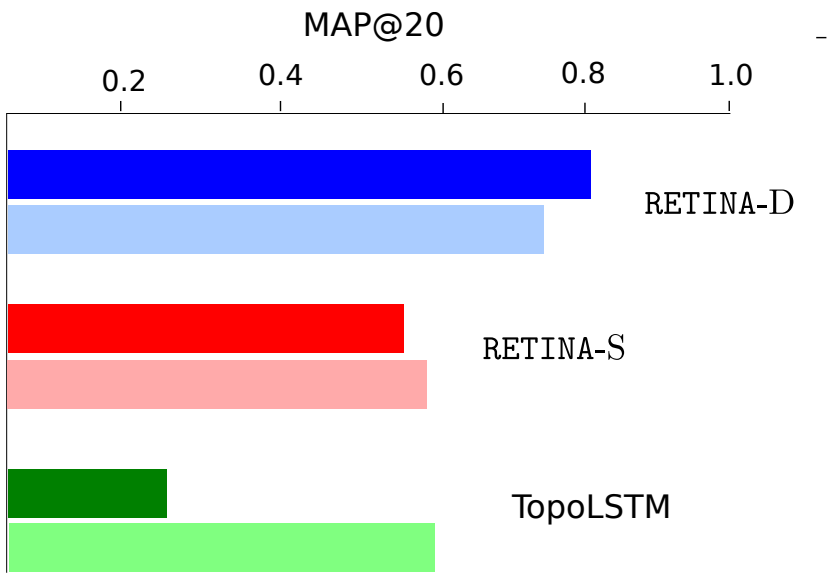

FIGURE 5.2: Comparison of RETINA in static (red; RETINA-S) and dynamic (blue; RETINA-D) setting with TopoLSTM (green) to predict potential retweeters when the root tweet is – hateful (dark shade) vs non-hate (lighter shade).

significant margin for all the evaluation metrics, TopoLSTM emerges as the best baseline in terms of both MAP@20 and HITS@20.

In Figure 5.1, we compare RETINA in static and dynamic setting with TopoLSTM in terms of HITS@k for different values of k. For smaller values of k, RETINA largely outperforms TopoLSTM, in both dynamic and static setting. However, with increasing k-values, the three models converge to very similar performances.

We find that the contribution of the exogenous signal plays a vital role in retweet prediction, much similar to our findings in Table 5.3 for predicting hate generation. With the exogenous attention component removed in static as well as dynamic settings (RETINA-S† and RETINA-D†, respectively, in Table 5.4), performance drops by a significant margin. However, the performance drop is more significant in RETINA-D† for ranking users according to retweet probability (MAP@k and HITS@k).

Figure 5.2 provides an important insight regarding the retweet diffusion modeling power of our proposed framework RETINA. Our best performing baseline, TopoLSTM largely fails to capture the different diffusion dynamics of hate speech in contrast to non-hate (MAP@20

0.59 for non-hate vs 0.43 for hate). On the other hand, RETINA achieves MAP@20 scores 0.80 and 0.74 in dynamic (0.54 and 0.56 in static) settings to predict the retweet dynamics for hate and non-hate contents, respectively. One can readily infer that, our well-curated feature design by incorporating hate signals along with the endogenous, exogenous, and topic-oriented influences empowers RETINA with this superior expressive power.

CHAPTER 6

Conclusion and Future Work

6.1 Conclusion

The majority of the existing studies on online hate speech focused on hate speech detection, with a very few seeking to analyze the diffusion dynamics of hate on large-scale information networks. We bring forth the very first attempt to predict the initiation and spread of hate speech on Twitter. Analyzing a large Twitter dataset that we crawled and manually annotated for hate speech, we identified multiple key factors (exogenous information, topic-affinity of the user, etc.) that govern the dissemination of hate. Based on the empirical observations, we developed multiple supervised models powered by rich feature representation to predict the probability of any given user tweeting something hateful. We proposed `RETINA`, a neural framework exploiting extra-Twitter information (in terms of news) with attention mechanism for predicting potential retweeters for any given tweet. Comparison with multiple state-of-the-art models for retweeter prediction revealed the superiority of `RETINA` in general as well as for predicting the spread of hateful content in particular.

6.2 Future Work

In this study, the mode of hate speech spread we primarily focused on is via retweeting, and therefore we restrict ourselves within textual hate. However, spreading hateful contents packaged by an image, a meme, or some invented slang are some new normal of this age and leave the space for further studies in the future.

Bibliography

[1] U. N. H. R. Council, 'Report of the Independent International Fact-Finding Mission on Myanmar', *A/HRC/39/64* **2018**.

[2] K. Müller, C. Schwarz, 'Fanning the flames of hate: Social media and hate crime', *Available at SSRN 3082972* **2019**.

[3] P. Fortuna, S. Nunes, 'A survey on automatic detection of hate speech in text', *ACM Computing Surveys (CSUR)* **2018**, *51*, 1–30.

[4] B. Ross, M. Rist, G. Carbonell, B. Cabrera, N. Kurowsky, M. Wojatzki, 'Measuring the Reliability of Hate Speech Annotations: The Case of the European Refugee Crisis', *CoRR* **2017**, *abs/1701.08118*.

[5] B. Mathew, R. Dutt, P. Goyal, A. Mukherjee in WebSci, **2019**, pp. 173–182.

[6] S. Dutta, S. Masud, S. Chakrabarti, T. Chakraborty, 'Deep Exogenous and Endogenous Influence Combination for Social Chatter Intensity Prediction', *arXiv* **2020**, *2006.07812*.

[7] S. A. Myers, C. Zhu, J. Leskovec in ACM SIGKDD, Beijing, China, **2012**, 33–41.

[8] Z. Waseem, D. Hovy in NAACL, **2016**, pp. 88–93.

[9] T. Davidson, D. Warmsley, M. W. Macy, I. Weber in ICWSM, **2017**, pp. 512–515.

[10] P. Badjatiya, S. Gupta, M. Gupta, V. Varma in WWW, Perth, Australia, **2017**, 759–760.

[11] P. Fortuna, J. Rocha da Silva, J. Soler-Company, L. Wanner, S. Nunes in Proceedings of the Third Workshop on Abusive Language Online, ACL, Florence, Italy, **2019**, pp. 94–104.

[12] A. Ghosh Chowdhury, A. Didolkar, R. Sawhney, R. R. Shah in ACL-SRW, **2019**, pp. 273–280.

[13] E. Fehn Unsvåg, B. Gambäck in Proceedings of the 2nd Workshop on Abusive Language Online (ALW2), Brussels, Belgium, **2018**, pp. 75–85.

[14] R. Kapoor, Y. Kumar, K. Rajput, R. R. Shah, P. Kumaraguru, R. Zimmermann, 'Mind Your Language: Abuse and Offense Detection for Code-Switched Languages', *arXiv* **2018**, *1809.08652*.

[15] H. Li, H. Li, S. S. Bhowmick in SIGMOD, Association for Computing Machinery, New York, NY, USA, **2020**, 1829–1840.

[16] W. O. Kermack, A. G. McKendrick, 'A contribution to the mathematical theory of epidemics', *Proceedings of the royal society of london. Series A Containing papers of a mathematical and physical character* **1927**, *115*, 700–721.

[17] C. Li, J. Ma, X. Guo, Q. Mei in WWW, Perth, Australia, **2017**, 577–586.

[18] Q. Cao, H. Shen, K. Cen, W. Ouyang, X. Cheng in CIKM, **2017**, 1149–1158.

[19] A. Khosla, A. Das Sarma, R. Hamid in WWW, Seoul, Korea, **2014**, 867–876.

[20] S. Bourigault, S. Lamprier, P. Gallinari in WSDM, **2016**, 573–582.

[21] S. Gao, H. Pang, P. Gallinari, J. Guo, N. Kato, 'A Novel Embedding Method for Information Diffusion Prediction in Social Network Big Data', *IEEE Transactions on Industrial Informatics* **2017**, *13*, 2097–2105.

[22] C. Yang, M. Sun, H. Liu, S. Han, Z. Liu, H. Luan, 'Neural Diffusion Model for Microscopic Cascade Prediction', *ArXiv* **2018**, *abs/1812.08933*.

[23] J. Wang, V. W. Zheng, Z. Liu, K. C. Chang in ICDM, **2017**, pp. 475–484.

[24] C. Yang, J. Tang, M. Sun, G. Cui, Z. Liu in IJCAI, **2019**, pp. 4033–4039.

[25] Z. Wang, W. Li in IJCAI, **2019**, pp. 3828–3834.

[26] F. Zhou, X. Xu, G. Trajcevski, K. Zhang, 'A Survey of Information Cascade Analysis: Models, Predictions and Recent Advances', *ArXiv* **2020**, *abs/2005.11041*.

[27] Y. Hu, S. Farnham, K. Talamadupula in ICWSM, **2015**, pp. 168–177.

[28] G. Wenzek, M.-A. Lachaux, A. Conneau, V. Chaudhary, F. Guzmán, A. Joulin, E. Grave, 'CCNet: Extracting High Quality Monolingual Datasets from Web Crawl Data', *ArXiv* **2020**, *abs/1911.00359*.

[29] M. Muhlmeyer, J. Huang, S. Agarwal, 'Event Triggered Social Media Chatter: A New Modeling Framework', *IEEE TCSS* **2019**, *6*, 197–207.

[30] A. De, S. Bhattacharya, N. Ganguly in WWW, Lyon, France, **2018**, 549–558.

[31] F. Hamborg, N. Meuschke, C. Breitinger, B. Gipp in ICIS, (Eds.: M. Gaede, V. Trkulja, V. Petra), Berlin, **2017**, pp. 218–223.

[32] A. Lajmanovich, J. A. Yorke, 'A deterministic model for gonorrhea in a nonhomogeneous population', *Mathematical Biosciences* **1976**, *28*, 221–236.

[33] Q. Le, T. Mikolov in ICML, **2014**, pp. 1188–1196.

[34] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, I. Polosukhin in NIPS, **2017**, pp. 5998–6008.

[35] T. Chen, C. Guestrin in SIGKDD, **2016**, pp. 785–794.

[36] D. P. Kingma, J. Ba in ICLR (Poster), **2015**.

[37] D. Kempe, J. Kleinberg, É. Tardos in SIGKDD, **2003**, pp. 137–146.

[38] W. Chen, Y. Yuan, L. Zhang in 2010 IEEE International Conference on Data Mining, **2010**, pp. 88–97.