EXAMINING UNTEMPERED SOCIAL MEDIA: INFORMATION MUTATION ON GAB.AI

by

Sai Eshwar Prasad Muppalla

A thesis submitted to the faculty of
The University of North Carolina at Charlotte
in partial fulfillment of the requirements
for the degree of Master of Science in
Computer Science

Charlotte

2018

Approved by:

Dr. Siddharth Krishnan

Dr. Bojan Cukic

Dr. Minwoo Lee

ABSTRACT

SAI ESHWAR PRASAD MUPPALLA. Examining Untempered Social Media: Information Mutation on GAB.ai(Under the direction of DR. SIDDHARTH KRISHNAN)

Online social media often mirrors the social phenomenon of "Chinese Whispers" where information mutates during dissemination. Many a time, Twitter and Facebook in conjunction with smaller "fringe" communities like Gab.ai, often echo perspectives to facts and many do so under the guise of "free speech". In this work, we propose a novel framework to examine the information mutation and in some cases online extremism using approximately 43 million posts (both original content and comments) from 450,000 users on Gab in conjunction with 3 million related blogs and news articles. We develop a system that mines the interaction between Gab and mainstream news media, when there is discourse commonality (for eg. during a shock event - say Unite the Right protests in Charlottesville during August 2017) and show evidence of how information mutates to the tune of the echoes within the Gab social system. To demonstrate our framework, we present two case studies of information mutation and online extremism, namely - the Charlottesville Unite the Right protests and the Pittsburgh synagogue shooting in October 2018. Particularly, in the context of the Pittsburgh shooting, we present a thorough analysis of content similar to that of the shooter and evolving narratives of distorted information.

DEDICATION

Dedicated to my family and friends.

ACKNOWLEDGEMENTS

First and foremost, I would like to thank my advisor, Dr. Siddharth Krishnan. I feel extremely privileged to be his first master's thesis student. I would like to extend my gratitude for giving me this opportunity and providing his interminable support and guidance through out the completion of my thesis. I appreciate all his contributions, time and ideas, to make my thesis experience productive.

I would also like to thank the rest of my thesis committee: Dr. Bojan Cukic and Dr. Minwoo Lee for their insightful feedback and encouragement. They played an integral role in inculcating and fostering the researcher in me. My sincere appreciation to the University of North Carolina at Charlotte for equipping me with necessary infrastructure and aiding me in the entire process.

This would have not been possible without consistent encouragement from my parents Narasimha Murthy MVL, Janaki Devi MB, my brothers Pavan Kumar and Shankar Nag and my friend Tejashree Hegdekatte. I would like to thank my fellow Network Analytics and Social Computing Lab (NASCL) members Arun Kumar Bagavathi and Gabriel Fair for their valuable time and support.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1: INTRODUCTION

Online fringe social media have become a fertile ground for individuals and groups to express unbridled opinion and galvanize supporters for their cause [1]. While most mainstream social media like Reddit, Twitter, Facebook, etc. moderate their content, in recent times the emergence of outlets like 4chan and Gab.ai have given extremist groups large content delivery networks to broadcast their message. The Gab social network was created in August 2016 and accrued tremendous support while leading up to the US Presidential elections, which resulted in a major political disruption - Donald Trump's victory. Primarily pitched as an unmoderated and censorship-free forum under the umbrella of "free speech", Gab morphed into an extremist echo chamber [2] garnering close to 450,000 users. Recently, the social network made headlines when it was revealed that Robert Bowers, the individual behind the Pittsburgh Synagogue shooting, was an active member of Gab and used it to express his anti-semitic views.

Forums like Gab and 4chan allow for vociferous vocalization of radical perspectives that can drown out accurate sources of information [3]. In recent times, we have observed that such discourse combined with exploitation of online social networks [4], is an effective strategy to recruit activists, instigate the public, and ultimately culminate in riots and protests as witnessed recently in Charlottesville or Portland.

By analyzing 43 million posts along with 3 million related news and blogs, this research focuses on analyzing how information mutates and morphs into *alternative narratives*. By developing a novel framework that studies the behavior of a social system like Gab during the outbreak of events, we are able to systematically characterize how there is a disagreement between the reporting of major news sites and the opin-

ions expressed in communities like Gab. Through the lens of two major events (Unite the Right Charlottesville protests and Pittsburgh Tree of Life Synagogue shooting), we demonstrate how Gab quickly attempts to disavow the news and attempt to form alternative narratives that echo the extremist principles that are an undercurrent of the social system.

1.1 Problem Statement

With the upswing in the usage of social media across the globe there is information exchange between social media and main stream news. In particularly Gab.ai is used as a platform to share alternative news and the news is spread rapidly taking its course in different direction over time.

In many ways alternative news media is dependent on platforms like gab.ai to spread its news to public, whereas social media like gab.ai - a free speech social network uses it to mutate the information. Such information from the news is spread rapidly and developed based on the popularity of the event.

The goal of the research is to propose a robust framework which unpacks different news chains from social media and expose how the information is mutated by the influence of alternative news media in a given time. Essentially we answer how the information from mainstream news media mutates in Gab.ai. Uncovering such story chains helps in understanding the type of information flow from traditional news media and it's dramatic appearance on social media.

1.2 Organization Of Thesis

This section briefly outlines the four main chapters of the thesis. The chapter 2 provides a detailed survey of different text clustering techniques, followed by Event detection methodologies in traditional mass and social media. Our work uses the the classical Online incremental clustering algorithm which is described under Unspecified event detection for sub-event detection. This chapter also discuses about

cosine similarity and jaccard coefficient measures used in our work. In chapter 3, a thorough review of the previous research and literature is presented. It summarizes the results and interpretations of previous work. The research framework is described in adequate detail in Chapter 4 through various phases. Each step involved in the procedure, starting from data pre-processing followed by sub event detection and story chaining has been explained. This chapter also comprises a comprehensive visual representation of the entire framework. A factual reporting of the study results is presented in chapter 5. The results encompasses a thorough discussion on case studies and their analysis.

CHAPTER 2: BACKGROUND

2.1 Text Clustering

Text clustering is one of the widely used technique in work related to text. We used different text clustering methods in our research to identify the sub-events in any given time frame. Here we present different clustering methods and their advantages in detail by studying the survey written by Aggarwal et al [5].

2.1.1 Feature Selection And Transformation Methods For Text Clustering

Clustering methods are dependent on the noise present in the features of the clustering process. Several commonly used words like "the" also called stop words are not much useful in augmenting the clustering quality. In addition to feature selection we also need to transform the features in order to increase the clustering quality. We have several techniques such as Latent Semantic Indexing(LSI), Probabilistic Latent semantic analysis (PLSA) and Non-negative Matrix Factorization.

2.1.1.1 Feature Selection Methods

Feature Selection methods are very commonly used and easy to apply for text categorization in supervised selection process. There are several unsupervised methods used in feature selection that are described below.

1. **Document Frequency-based Selection** This method used the frequency of the document to filter the features that are not relevant in clustering. The words also known as stop words which are too frequent such as "a", "an", "the", "of" can be removed which are very common. In general there are nearly 300 to 400 set of such stop words. Also we remove words which are extremely infrequent because they don't help in similarity computations used in clustering methods.

TF-IDF method also removes most common words in a smooth way.

2. **Term Strength** It is used as stop word removal by measuring how informative a word is in identifying two documents. For any two documents x and , the term strength s(t) is the probability:

$$s(t) = P\big(t\epsilon y | t\epsilon x\big). \tag{2.1}$$

Two documents are related to each other if their cosine similarity is above certain threshold defined by user. Then the term strength s(t) is defined by random sampling of number of pairs of related documents:

$$s(t) = \frac{\text{Number of pairs in which t occurs in both}}{\text{Number of pairs in which t occurs in the first of the pair}}. \tag{2.2}$$

3. **Entropy-based Ranking** The quality of the word is measured by the entropy change when the word is removed. The entropy E(t) of a word t in n documents is as follows:

$$E(t) = -\sum_{i=1}^{n}\sum_{j=1}^{n}\left(S_{ij}\cdot\log\left(S_{ij}\right) + \left(1-S_{ij}\right)\cdot\log\left(1-S_{ij}\right)\right). \tag{2.3}$$

Here $S_{ij} \in (0,1)$ is the similarity between ith and jth document in n, after the word t is removed:

$$S_{ij} = 2^{-\frac{dist(i,j)}{dist}}, \tag{2.4}$$

where $dist(i,j)$ is the distance between i and j after t is removed and dist is the average distance between documents after word t is removed.

4. **Term Contribution** This concept considers the fact that text clustering is based on document similarity. Therefore the contribution of the word to the similarity of documents is the product of normalized frequencies in both the documents.

2.1.1.2 LSI-Based Methods

Method such as Latent Semantic Indexing (LSI) is based on dimensional reduction in which each feature is linearly aligned with the features in the data. LSI is closely related to problem of Principal Component Analysis(PCA) or Singular Value

Decomposition (SVD). For a d-dimensional data set, PCA constructs the symmetric $d \times d$ covariance matrix C of the data, (i, j)th entry is the covariance between i and j.

$$C = P \cdot D \cdot P^T \tag{2.5}$$

Concept Decomposition using clustering uses the clustering itself as the dimensionality reduction technique.It uses any clustering technique on the documents. The terms which are frequent in the centroids of the clusters are used as basis vectors orthogonal to each other. This helps in enhancing the clusters and hence can be applied on the reduced representation.

2.1.1.3 Non-Negative Matrix Factorization

The non-negative matrix factorization (NMF) technique is a clustering technique uses latent space method.Here the vectors correspond to topics and hence if the document belongs to a cluster can be determined by the largest component of any vector. The coordinate of the document vector is non-negative. Let A be the $n \times d$ term document matrix. For k clusters ,the non-negative matrix factorization method determines the matrices U and V which minimize the objective function:

$$J = (1/2) \cdot \left\| A - U \cdot V^T \right\|. \tag{2.6}$$

An interesting observation of Matrix factorization is that it determines the word clusters rather than document clusters. This technique is also equivalent to spectral clustering.

2.1.2 Distance-based Clustering Algorithms

Distance-based Clustering Algorithms uses similarity functions to meseaure how closely two objects are related. The most well known function is cosine similarity function. Let $U = (f(u_1) \ldots f(u_k))$ and $V = (f(v_1) \ldots f(v_k))$ be the damped and normalized frequency term vector in two different documents U and V . The values $u_1 \ldots u_k$ and $v_1 \ldots v_k$ represent the normalized term frequencies, and the function

$f(\cdot)$ represents the damping function. The damping functions for $f(\cdot)$ could be the square-root or the logarithm. Then, the cosine similarity between the two documents is:

$$\text{cosine}(U, V) = \frac{\sum_{i=1}^{k} f(u_i) \cdot f(v_i)}{\sqrt{\sum_{i=1}^{k} f(u_i)^2} \cdot \sqrt{\sum_{i=1}^{k} f(v_i)^2}}. \tag{2.7}$$

2.1.2.1 Agglomerative And Hierarchical Clustering Algorithms

Hierarchical clustering algorithms are used for multidimensional, categorical and textual data. Agglomerative hierarchical algorithms are used as searching methods because of its structure being in the form of tree and can be used in search process. It increases the efficiency of the search. In agglomerative clustering we merge the documents into clusters by checking the similarity with each other. These hierarchical clustering algorithms further merge the groups again based on the similarity between them. This process of agglomerating the documents into clusters forms hierarchy in which the internal nodes are the merged cluster groups and the leaf nodes are the documents. Each node is formed by its two children nodes by merging into one. There are various methods such as:

1. **Single Linkage Clustering** : In this type of clustering, the similarity is obtained by getting the greatest similarity between two groups of documents. Here we merge two two groups based on the similarity of the closet pair of documents of one group to the closest pair of documents in any other pair. It is easy to implement because we first find all similarity pairs, sort them and merge them successively.
2. **Group-Average Linkage Clustering**: In this type of clustering, the similarity is computed based on the average similarity between the pairwise documents in different clusters. This is clearly slower than the single-linkage Clustering because of the computation needed for the average similarity of large pairs.
3. **Complete Linkage Clustering**: Complete linkage clustering takes the simi-

larity between any two clusters as the worst-case similarity between any pair of documents in the clusters. It avoids chaining by placing any pair of disparate points in the same cluster.

2.1.2.2 Distance-Based Partitioning Algorithms

Partitioning algorithms are used in to create cluster of objects. We will discuss two distance based clustering.

k-medoid clustering algorithms: In this clustering algorithm, set of points are chosen as the anchor points around which clusters are built. These are the medoids. The algorithm is used to determine a set of documents form the corpus around which the clusters are formed. Each document is assigned to the closest from the collection. This will form a set of clusters which are improved by a randomized process. It sis an iterative process in which set of k values are used as representative clusters successively by randomized inter-changes. In each iteration the randomly picked representative in current set is replaced with randomly picked from collection if the objective function is improved until convergence. The main advantage of k-medoids clustering is used for text data and the disadvantage is it requires large number of iterations for he convergence. It won't work well for sparse data such as text.

k-means clustering algorithms : The k-means clustering algorithm also uses k representatives around which clusters are formed. It is started with k seeds from the corpus and each document is compared with these seeds and assigns to these seeds based on the similarity. In the next iteration each seed is replaced by its centroid formed due to new cluster. It is done until the convergence and this convergence need only smaller number of iterations than the k-medoids clustering algorithm. The main disadvantage is the centroid may contain large number of words and hence will slow down the calculation of similarities.

2.1.2.3 A Hybrid Approach: The Scatter-Gather Method

This approach uses both hierarchical and partitional clustering algorithms. It uses the hierarchical clustering algorithm to the find the initial set of seeds. These seeds are used in k-menas clustering algorithm to obtain good clusters. Two methods are used to create initial set of seeds, such as buckshot and fractionation.

Buckshot: Consider k, the number of clusters that needs to be formed and n as the number of documents. The buckshot picks the seeds by an overestimate $\sqrt{k \cdot n}$ of the seeds and then agglomerates to k seeds. Standard agglomerative hierarchical clustering algorithms are applied to this initial sample of $\sqrt{k \cdot n}$ seeds.

Fractionation: This algorithm breaks the corpus into n/m buckets of size $m > k$. Now an agglomerative algorithm is used on these each buckets to reduce them by v. Hence we will have total of $\nu \cdot n$ points at the end. And the process is repeated by treating each of these points as an individual record.

Split Operation: In order to refine the cluster into further granularity we use split operation technique. We use buckshot procedure mentioned above on each document in a a cluster for $k = 2$ and forming new clusters with new centres. This requires $O(k \cdot n_i)$ for a cluster with n_i data points and hence splitting takes $O(k \cdot n)$.

Join Operation: In this operation it merges similar clusters into a single cluster. The merge operation is performed by computing the topical words for each cluster by taking the most frequent words of the centroid. Similarity is measured by taking the intersection between the topical words of two clusters.

2.1.3 Word And Phrase-Based Clustering

In finding the clusters of documents we can see the problem of high-dimensional domain of text documents in two ways. In We define the term-document matrix as an $\pi \times d$ matrix, where n is the number of documents and d is the number of terms in which $(i, j)th$ is the frequency of jth them in ith document. Since the matrix has

very few number of words this is a sparse matrix. Clustering the rows is clustering the documents and clustering the columns in the matrix is clustering the words.

2.1.3.1 Clustering With Frequent Word Patterns

This approach is based on the frequent pattern mining algorithms. Here we are dealing documents rather than transactions. It is mainly used to cluster the low dimensional frequent term sets as cluster candidates. Frequent Set is a cluster which corresponds to all documents which has that frequent term set. Let us now consider R as the frequent set terms as clustering. f_i is the number of frequent term sets in R of ith document. The f_i is initialized to one at-least for the complete coverage and to minimize the overlap it should be a slow as possible. So the average value of $(f_i - 1)$ is a slow as possible for the documents in the cluster. The entropy overlap is the sum of values of $-(1/f_i) \cdot \log(1/f_i)$ for all documents in the cluster.

2.1.3.2 Leveraging Word Clusters For Document Clusters

In this there are two phases for document clustering. In the first phase, the mutual information between words and documents is preserved to determine word-clusters from the documents. The second phase determines the condensed representation(word-clusters) of the documents. In performing the document clustering word occurrences are replaced with word-clusters.

Consider $X = x_1 \dots x_n$ random variables which corresponds to rows(documents) and $Y = y_1 \dots y_d$ random variables corresponds to column(words). Partition X into k clusters and Y into 1 clusters. Clusters are denoted as $\hat{X} = \hat{x}_1 \dots \hat{x}_k$ and $\hat{Y} = \hat{y}_1 \dots \hat{y}_{lk}$ We want to find the maps $\overline{C}_X$ and $\overline{C}_Y$ that define the clustering.

$$C_X : x_1 \dots x_n \Rightarrow \hat{x_1} \dots \hat{x_k}, \tag{2.8}$$

$$C_Y : y_1 \dots y_d \Rightarrow \hat{y_1} \dots \hat{y_l}. \tag{2.9}$$

2.1.3.3 Clustering With Frequent Phrases

This method of text clustering treats each document as a string rather than treating the document as a bag of words. Especially each document is treated as string words than characters. In bad of words representation it maintains the order of the words but not in case of string words. This method uses indexing technique to organize the phrases in the document to create the clusters. Following are the steps in creating clusters.

1. The first step performs the cleaning of strings representing the documents. Different stemming algorithms can be used to remove prefix and suffix of words and convert a plural word to singular word. Each end of the sentence is marked and which are not words are stripped.
2. The second step is to form base clusters.A suffix tree is used in representing the frequent phases which contain the suffixes of entire collection. Each node of the suffix tree is a group of documents and hence it is a base clustering. Each base cluster is given a score computer as the product of number of documents in the cluster and a non-decreasing function of the length of the underlying phrase.
3. The suffix tree generated in previous step does not have strict partitioning and therefore have overlaps. the third step merges the clusters based on the similarity of the sets. Let P and Q be the documents sets for tow clusters. The similarity is defined as :

$$BS(P,Q) = \left\lfloor \frac{|P \cap Q|}{\max\{|P|, |Q|\}} + 0.5 \right\rfloor . \tag{2.10}$$

2.1.4 Probabilistic Document Clustering And Topic Models

A method for Probabilistic document clustering is that of *topic modeling*. Topic modeling is nothing but creating the probabilistic model for text documents. The corpus in the probabilistic document clustering is represent as function of hidden random variables for which the parameters are estimated using the documents. There

are two methods for topic modeling, such as Probabilistic Latent Semantic Indexing and Latent Dirichlet Allocation(LDA).

We will describe the probabilistic latent semantic indexing method here. A set of random variables $P(T_j|D_i)$ and $P(t_l|T_j)$ model the probability of term t_l occurring a document D_l. The probability $P(t_l|D_i)$ of the term t_l occurring Document D_i can be expressed as below:

$$P(t_l|D_i) = \sum_{j=1}^{k} p(t_l|T_j) \cdot P(T_j|D_i). \tag{2.11}$$

Each term t_l and document D_i, we can generate a $n \times d$ matrix of probabilities, where n is the number of documents and d is the number of terms. The constrained optimization problem optimizes the value of the log likelihood probability $\sum_{i,l} X(i,l) \cdot \log(P(t_l|D_i))$ such that the probability values of topic-document and term-topic spaces sum to 1: subject to the constraints that the probability values over each of the topic-document and term-topic spaces must sum to 1:

$$\sum_{l} P(t_l|T_j) = 1 \quad \forall T_j, \tag{2.12}$$

$$\sum_{j} P(T_j|D_i) = 1 \quad \forall D_i \tag{2.13}$$

for each entry (j , i) in P_1 do update

$P_1(j,i) \leftarrow P_1(j,i) \cdot \sum_{r=1}^{d} P_2(r,j) \cdot \frac{X(i,r)}{\sum_{v=1}^{k} P_1(v,i) \cdot P_2(r,v)}$

Normalize each column of P1 to sum to 1;

for each entry (l,j) in P_2 do update

$P_2(l,j) \leftarrow P_2(l,j) \cdot \sum_{q=1}^{n} P_1(j,q) \cdot \frac{X(q,l)}{\sum_{v=1}^{k} P_1(v,q) \cdot P_2(l,v)}$

Normalize each column of P2 to sum to 1;

2.1.5 Online Clustering With Text Streams

Streaming text clustering is challenging because of the fact that text needs to be maintained continuously. One of the method $Online Spherical k-Means Algorithm (OSKM)$, divides the streams into smaller segments, where each segment is processed. Then K-means is applied to each data segment to cluster them. A phrasal clarification in

improving the quality of cluster known as *semantic smoothing* because it reduces the noise with semantic ambiguity.

Another approach works by modelling the soft probability $p(w|C_j)$ for a word w and for a cluster C_j. The probability $p(w|C_j)$ is combination of (a) maximum likelihood model that computes the probabilities for words of a cluster (b) A translated probability which determines the maximum probability of each word for a topic. $p(w|C_j)$ is used to estimate $p(d|C_j)$ by using the product of consecutive words in the document. Hence $f(w, d)$ is used for a word w and document d.

$$p(d|C_j) = \prod_{w \in d} p(w|C_j)^{f(w,d)}. \tag{2.14}$$

2.2 Event Detection

2.2.1 Event Detection In traditional Mass Media

This section provides a summary of event detection techniques in conventional media outlets. It can be broadly classified as document pivot i.e. based on document features and feature pivot i.e. based on temporal feature techniques.

2.2.1.1 Document Pivot Methodology

As a part of Defense Advanced Research Projects Agency initiation, event detection was a part of TDT (Topic Detection and Tracking) program [6] which pertaining to organization of event based textual news document streams. TDT programs motivation was to keep users updated about news development by providing the most important technology for news monitoring tools taken from various traditional media resources like newswire, broadcast news etc. It majorly consisted of segmentation, detection and tracking tasks. The aim of performing these tasks is to divide the news text into meaningful straight-line story, detect unforeseen events, and then to track the changes in formerly reported event.

In general event detection consists of three phases namely data preprocessing, data representation and data organization ([7], [8]). Data pre-processing involves removing

the stopwords from the document. Stopwords filters out insignificant words like the, is, are etc. from the text document. Further stemming and tokenization methods are applied to the document to remove affixes of a word to get the root word. Data representation pertains to creating bag of words(vectors) which adds the corresponding entries that appear in the document. The term frequency-inverse document i.e. tf-idf technique [9] is used to determine the importance of a word to the text document. But it has its drawbacks like curse of dimensionality for the vector model when the text document is lengthy. The model may not be able to detect the similarity or differences in the events as the vectors discard the order of words, semantic and the syntactic features of the words in the text document.

This led to the exploration of other ways to perform data representation like entity vector [10] which aims to pull information based on questions like who, what, when and where [11]. Mixed vectors ([12], [10]) are based on integration of term and entity vectors. Probabilistic frameworks consisting of content and time information [13] is the probabilistic representation [14] approach which includes language models. Traditional metrics like Euclidean distance, Pearson's correlation coefficient, and cosine similarity can be used to check how the events are similar to each other. Recent measures proposed are Hellinger distance [15] and the clustering index [16].

TDT is classified into Retrospective event detection (RED) and New event detection (NED) ([7] , [17]). RED aims to unearth previously unidentified events from collected historical data [7], whereas NED focuses on real time streams to detect new events [17]. Both these techniques use Clustering based algorithms ([18], [19]). RED uses iterative clustering like hierachial agglomerative clustering (HAC) [20], which is a bottom up approach which takes the entire document and organizes it into topic clusters. HAC considers each data point in a cluster and the closest clusters are merged together and this is stopped when certain criteria are met or if all data points are merged into a cluster. Variations of HAC are employed to detect new events for

TDT tasks like using a two-layer HAC method to decrease false positives which is based on affinity propagation [21]. The variations of k-means clustering like k-median method and k-means++ methods have also been proposed [22]. According to [17] the new event detection is said to be query free when it comes to retrieval tasks as there are no prior event information and thus it cannot be queried. NED applies incremental(greedy) clustering algorithm methods as this approach should provide the decision on events (old or new) as soon as the text documents are streamed. The incremental methods process the stream inputs and then merges events that is similar to each other or creates a new cluster if they exceed the threshold for the similarity measure [17]. The cons of using this methodology is that it is both time and resource intensive and may need additional system efficiency [23] to make it feasible. The resource requirements can be improvised by using a sliding time window over the previous stories and then comparing the new story based on the most recent number of stories ([7], [24], [23]). Here the assumption is that the stories that are related to the same event lie the same time frame. Other techniques employed for increasing the system efficiency is to put limitations on the number of terms per document and number of total terms kept, also parallel processing can be applied [23].

These above specified techniques which use clustering documents based on the text similarity for event detection are called document-pivot techniques. But this technique is not very useful for events hidden in large amounts of noisy textual stream data ([25], [26], [27], [28]). This technique also falls short when it comes to handling the speed and scale of social media data.

2.2.1.2 Feature-Pivot Techniques

Detecting trends over large collection of data collection involves identifying the topic areas which were missed previously or are currently gaining importance within the corpus [29]. In the recent times there has is significance for the bursty event detection techniques in traditional media ([30], [31], [32], [33] , [34], [35]). The

technique of feature pivot model in event detection considers the event in the text stream as a bursty activity which some features rising sharply as the event occurs at a certain frequency. Hence the event is depicted by several keywords which shows the bursty appearance counts [30]. Here the underlying assumption is that some of the words that are related to each other have an increased usage as the event takes place. The difference from the RED and NED document pivot approach is that these techniques check and analyze the feature distribution and then the events are detected by grouping bursty features that have same trends. In his formative work, [30] proposed an infinite-state automaton to model the document arrival times in a text stream to detect the bursts that display high intensity over short periods of time. The frequencies of the individual words are corelated to the probabilistic automaton state and the bursts are captured by state transitions which implies a significant change the frequency of the word. [31] proposed a word appearance model as binomial distribution, set a heuristic-based threshold to identify the bursty words and grouped the bursty features to detect the bursty events. [33] used spectral analysis for detecting various event characteristics by using discrete Fourier transformation (DFT) to categorize the features. The signals from the time domain are converted to frequency domain by the help of DFT. Here the disadvantage is that DFT cannot determine the time frame of the bursty activity. Hence, [32] applied Gaussian mixture models which also identifies the periods associated with the feature bursts. For online detection, statistically significant tests of n-gram word frequency with a time period was proposed by [36] for detection of events in streaming news by using an incremental suffix tree data structure to reduce the time and space complexities.

2.2.2 Event Detection In Twitter

The event detection can be classified based on whether there is an information available on the event of interest as specified and unspecified techniques. Unspecified event detection depends on the temporal signal of the textual streams to identify

the real-world event occurrence as it has no prior knowledge of the event available. Specified event on the other hand depends on information features specific to the event like the venue, time, type and description. In the below discussion the words twitter, and social media have been used interchangeably.

2.2.2.1 Unspecified Event Detection

The unknown events of interests on social media platform are usually emerging events, breaking news, and general topics that attract the attention of many users and since the online posts reflect the events as they are unfolding, they are a very convenient for unknown event detection. A sudden increase in usage of some specific keywords leads to new events of interest as they display a burst of features in the social media stream. These kind of bursty features that repeat can be then bifurcated as trends [37]. Social media posts also have endogenous trends in abundance [38]. Hence the techniques involved in unspecified event detection must incorporate scalable and efficient algorithms to differentiate between the new events of interest from nonevent trends. One such technique is proposed by [39] called TwitterStand which a news processing system based on Twitter data.

This technique captures tweets that correspond to late breaking news. Here, naive Bayes classifier is used to separate news from trivial information and then clusters of news are based on online clustering algorithm that use tf-idf and cosine similarity. Hashtags have been used to reduce the clustering errors. Also, time information is associated with cluster management and to deduce the clusters of interest. Another method that is presented by [40] collects, groups, ranks and tracks the breaking news from twitter data. The tweets are first sampled based on search queries that are predefined like hashtags "breakingNews" as keywords and then the content of these tweets is indexed with Apache Lucene.

A news story is formed based on similar messages and this similarity is calculated using tf-idf with higher weights for proper noun terms, hashtags, and usernames. The

Stanford Named Entity Recognizer (NER) is used to identify the proper nouns. A weighted combination of reliability based on number of followers and retweeted messages are used with time adjustment to determine the freshness of that message to rank the cluster. A new message is added to the cluster if they are similar to the first message and to the top-k terms in that cluster. The similarity comparison between the messages is improved by detecting the proper nouns and this in turn increase the system accuracy. Hot-streams is an application that is developed based on this technique.

The online NED technique is adapted by [41] for newsmedia [42], using the cosine similarity between the documents to check for new events that has not previously occurred in the tweets. The technique proposes an algorithm with constant time and space approach that is based on the adapted version of [43]âs locality sensitive hashing methods. This technique does not consider replies, retweets, hashtags and if the detected new events are nonevents or not. The results show that it is better to rank based on the number of users than the number of tweets and also if the entropy of the message is considered it reduces the spam messages in the output.

An online clustering method that clusters similar tweets and later classifies the content of the clusters into trivial events of events of interest where the focus here is on online detection of real-world events. Twitter- centric topics are cumbersome to detect as they are trending on Twitter, but they do not concentrate on real world event content [38]. Classical incremental clustering algorithms that are based on a threshold are used for clustering. Every message is tf-idf weighted vector of the content and the cosine similarity is used to calculate the distance between the message and the cluster centroids. Here, along with the preprocessing techniques like stop word removal and stemming, the hashtag term weight is doubled. Temporal, social, topical and twitter centric features are taken into consideration. The features are periodically updated in the clusters to form new clusters. Later a Support Vector

Machine (SVM) classifier that is trained on a labeled cluster with a set of features is used to determine if the input cluster contains real-world event content.

A clustering approach which is based on certain features of micro blog data is proposed by [44]. Here the features are based on 'topical words' that are pulled from the messages based on word frequency, entropy, occurrence of the word in the hashtags. A co- occurrence graph is created based on topical words that recur and top-down hierarchical clustering technique is applied to divide the topical words into clusters of events. Event chains are created based on maximum-weighted bipartite graph matching that can track changes among events occurring at different times. Lastly, the top-k events that outlines an event are created using cosine similarity with time intervals between the messages. The authors found that top-down divisive clustering algorithms works better in comparison to k-means and hierarchical clustering algorithms when it comes to unspecified event detection.

Cordeiro (2012) [45] proposed a continuous wavelet transformation considering the occurrences of hashtags along with the topic model inference combination using latent Dirichlet allocation (LDA) [46]. Here the hashtags are used to build the wavelets instead of individual words. A sudden increase in the count of the hashtag in consideration depicts an indicator that the event is of interest for that time period. Hashtags were retrieved from twitter data and then divided into groups of 5-minute intervals. The hashtags are built over a period of time by accounting the hashtag mentions in every 5-minute interval, and then linking all the tweets that refers to the particular hashtag. Techniques like peak and local maxima are used to identify the changes and the peaks in the hashtag signal. An improved summarization of the event description is obtained by applying LDA to all the tweets mentioning the hashtags in the corresponding time series.

2.2.3 Specified Event Detection

In specified event detection consists of familiar or social events that are scheduled to happen. The events are mostly specified in a complete or partial manner based on the content or the metadata information like the time, venue, location etc. Techniques involving machine learning, text mining and analysis are applied to exploit the Twitter textual data or metadata information.

Popescu and Pennacchiotti (2010) [47] aimed to detect events that evoke public discussions around controversial topics like politics or celebrities that leads to opposing outlooks in Twitter data. A detection framework is constructed that includes target entity (e.g., Narendra Modi), a given period (e.g., a certain day in a year), and a set of tweets about the target entity from the defined period of time and this forms the Twitter snapshot. A supervised gradient boosted decision tree [48] that is trained on a set of labeled data is used to distinguish events of interest from non-trivial events when a set of Twitter snapshots are input. The event snapshots are ranked based on a controversy model that allocates higher scores to event snapshots that are controversial that applies regression algorithms to a large number of features. The features are specific characteristics of Twitter data that entails linguistic, structural, buzziness, sentiment, and controversy features. The external features like web-news and news controversies are also included that captures entities that are likely to relate to real-world events. An additional feature is proposed to the controversy model that merges both the detection and scoring stages into a single system which yields improved performance. Hashtags acts as an important feature for the tweets data as it helps identify the tweet topic and determines the topical cohesiveness of a set of tweets.

In a future work, [49] used the same framework described above with other features that help detect description of events from Twitter. The importance and the number of the entities to apprehend a sense about the events of interest from the non-trivial

events is the aim of this proposal. Inspired from the document aboutness system [50] ranks the entities based on relative positional information, term-level information and snapshot-level information in a given snapshot with respect to their relative importance in that snapshot. Tools based on opinion extraction such as off-the-shelf part-of-speech (POS) tagger is used to improve the event and entity extraction.

A novel approach proposed by [51] for identifying concert events in Twitter data uses a factor graph model that simultaneously analyses discrete tweets and then it is clustered based on event type. A canonical value is induced for every property of the event. [52] collected geotagged Twitter data and preprocessed it for a particular region for a long period of time [53] to propose a a geosocial local event detection system to identify local festivals based on modeling the crowd behaviors over Twitter. This region is then categorized into various regions of interest (ROI) by applying k-means algorithm to the geographical coordinates (longitudes/latitudes). Then three main features are considered from historical data namely the number of tweets, users, and moving users within an ROI which is used to estimate geographical regularities of crowd within each ROI. A 6-hour time interval is used to form the estimation of the crowd behavior in each ROI. Then finally, statistics are compared from new tweets with the estimated behavior to detect unusual events in the particular geographical area. The authors realized better indicator of the local festivals was by an increased user activity in combination with an increase in the number of tweets.

Sakaki et al. (2010) [54] devised a classification problem for detecting events of specific nature like earthquakes where they trained an SVM on a labeled twitter data consisting of positive events (earthquakes and typhoons) and negative events (other events or nonevents). The features taken into consideration are the number of words (statistical), the tweet message keywords and the contextual user queries. This analysis of the number of tweets over a period of time helped discover an exponential distribution of events. Here the authors also employed Kalman filtering and particle

filtering [55] to estimate the earthquake center and typhoon trajectory based on the temporal and spatial information from Twitter.

Massoudi et al.(2011) [56] proposed a model to extract independent messages from microblogs which is based on a generative language modeling approach that uses query expansion and quality indicators. Here, the local frequency of query term is not considered by the authors. It includes âcredibility Indicatorsâ proposed by [57] that are the quality indicators such as emoticons, post length, shouting, capitalization and retweets and popularity (based on twitter followers). It also includes recency factor that is a difference between the query and the microblog post time. An average of the microblog-specific indicators is calculated into a single word and the prior probability of the microblog post is computed by adding the weight of the credibility indicators. The top-k terms are selected based on the query expansion technique that is seen in the user defined posts which is near to the query date which comprises of original and expanded query.

2.2.4 New Versus Retrospective Event

The Twitter data can be categorized as Retrospective event detection (RED) and New event detection (NED) as it was done in the traditional media based on the task, requirements of the application and also the event type. The NED techniques can be applied to detect unspecified real-world events like breaking news as this technique involves the monitoring of Twitter signals to discover near future new events. Sometimes, instead of the actual event the name, comment or a person which is related to the real-world event may end up becoming a trending topic on twitter. NED techniques can also be employed in case of specific events when there is monitoring task like news involving celebrities or any disaster occurrence etc. Sakaki et al [54] application of filtering techniques or the Popescu et al [47] exploitation of the extra added features could be integrated into the NED system to help focus on the general event of interest. NED techniques can also be employed to analyze previous events.

CHAPTER 3: RELATED WORK

In this section, we review related work on the impacts of social media on information framing. The rise of alternative news sources and their use to shape or frame news and information has been the subject of many recent studies([58], [59], [60]). Mass shooting events are frequently a subject of alternative interpretations of the news, converging around a small set of themes used for alternative interpretation, while giving the impression of a diversity of sources and support [3]. Recent studies have highlighted the advent of a social network, Gab.ai which was found to be very popular with alt-right users sharing a diversity of sites promoting alternative news framing ([61], [1], [62], [2]). While online extremism has increased ([63], [64]) so has the number of hate crimes committed in the United States. Many works focused on detecting online extremists and people who are promoting them in the social media like Twitter ([65], [66]) using machine learning methods by using text based features that people use in their posts/tweets.

One of the state of art papers in understanding the information flow between news media and social media is "Uncovering News-Twitter Reciprocity via Interaction Patterns" [67]. In this paper they explained the dependencies between traditional news media and social media and their interaction pattern. In addition, they also presented the rate at which interaction takes place between news and twitter. The framework they designed has the following components: Story chaining of news articles, retrieval of tweets related to news, identifying interaction patterns between news and tweets, clustering of interaction patterns and topic modeling.The story chaining algorithm chains the news articles based on the weighted scores of similarities using textual features, spatial(such as locations and geographical coordinates) features and actors

such as persons and organizations.The interaction pattern they classified between news article and twitter activity has different states encoded resulting in different sequence of states. They identify the source of information based on these interaction patterns for every news chain and form distinct clusters. These clusters are further processed to find the dissimilarities in in the news articles of the cluster. By this they find the direction of information flow over time between traditional media and Twitter. By the experiments they found Twitter is a medium to grab public attention on social events whereas news media reports events regarding political, economical and business articles.

Another interesting paper is "Uncovering Topic Dynamics of Social Media and News: The Case of Ferguson" [68]. It focuses on understanding the dynamics of news and social media and their relationship around the news events. They proposed a Single topic LDA (ST-LDA) that produces various topics for a news document and a single topic for tweets. They found discovery of topic improves by removing the noisy topics in news and tweets. Using this algorithm they studied a case- unrest in Ferguson shooting finding the dynamics of tweets and their differences and relationships with news. This paper contributes the technical problem to construct topic models for short and long texts. Their model (ST-LDA) takes all the words as a single tweet and label a topic to the tweet. The output of the ST-LDA is used to discover the temporal change in topics using a sliding window of one day. The results presented is able to detect the common topics in both news and tweets and label it to main topic.

In the paper "Information Contagion: an Empirical Study of the Spread of News on Digg and Twitter Social Networks"[69], they show how social networks plays a vital role in spread of information on news sites and how the structure of network influences the information flow. They considered the users social network data of Digg and twitter and found how news stories disseminates. They first characterize the structure of networks on Digg and Twitter by considering the number of fans

on each network exhibits. And they study the information spread by a measure of number of in-network votes a story receives and discover the dynamics of information spread. They claim the stories are spread farther in twitter and faster in Digg.

There is a tremendous amount of work done in the area of story chaining, one such work is "Analyzing Evolving Stories in News Articles"[70]. In this paper, they present an algorithm that detects the origin of event and groups the news articles by the order of time and segregate the news articles based on the soft probability which evolves into a story. They also proposed a method in the evolution of the concepts in a given set of documents. The paper includes a case study which demonstrate the scope to predict the future states of a story evolution by taking the news chains generated. Hossain et al [71] presented an automatic approach information discovery from the PubMEd abstracts. They describe an algorithm which automatically identifies the sequence of publications such that any two neighbouring publication have similarity in content. They also demonstrated the design of coherency of a story from one publication to other. The results demonstrated through the pipeline helps in minimizing thousands of documents to several hundreds of stories. Since this approach is an unsupervised method, Schlachter et al [72] describes methods to identify most coherent and meaningful story chains. They present two topic based models, the first measures the wellness of the story formed from the corpus at any given time and the second measures the story chain by topic consistency and its persistence. The former is done by the similarity comparison of the topics in a story chain and those expressed in the corpus. They considered that stories with similar topics will convey similar story of central corpus. They have come up with four categories to predict the story chain (1) very clear narrative, (2) somewhat clear narrative, (3) somewhat unclear narrative, (4) very unclear narrative. Their results indicate using topic model is an interesting aspect in accessing the narrative structure.

In addition, the paper "Connecting the Dots Between News Articles"[73] also in-

vestigates the methodology to automatically connect the dots such that it is easy to find the connections within news articles. The algorithm they provided connect two fixed end points by the mechanism of user feedback into their framework. They first formalized the characteristics of a story coherence and the influence with no link structure and followed by connecting the dots while maximizing the coherence through feedback and interaction.

Lastly there has been a few studies of how the the specifics of a news story diffuse across user's digital news feeds. Known as diffusion theory, proposed ways of studying this include measuring news incoherence, similarity, overlap, uniformity [74]. Or through the use of topic dynamics with Single Topic LDA [75]. And topic detection as proposed by the Normalized Mutual Information (NMI) framework [76]. Ning et al [67] proposed a framework for understanding the interaction patterns created by the flow of information between news and social media through the use of story chain modeling. These interaction patterns with Gab.ai and news sources are the target of our information mutation framework.

CHAPTER 4: METHODOLOGY

4.1 Dataset Description

Gab.com/Gab.ai is a social media forum, founded in 2016, as a social network that is committed to protecting free speech. Even though the description of the forum looks very similar to other social media networks like Twitter, Gab.ai is known for supporting individual liberty and free speech in online media[1]. Further, Gab.ai has strong restriction policies on posts and users promoting pornography, terrorism and violence. Users of gab.ai can share information via *posts*, *post replies*, and *reshares*. Figure 4.1 gives a timeseries plot for the frequency of posts, replies, and quotes appeared in Gab between August 2016 and October 2018.

We have a comprehensive collection of gab.com data with about **43 million** posts, replies, quotes posted between the date range of August 2016 and October 2018, about **15,000** groups and about user information of about **450,000** public users. It is evidential from Figure 4.1 that our dataset comprise of 55% posts, 30% replies, and 15% reshares. Figure 4.2 gives a distribution on participation level of all user types in our data. Gab provides choices for users to be one among the following user types:

- **Donor**: Users who donates to Gab to support its free speech movement
- **Investor**: Users who invests on Gab and receives perks from the Gab team
- **Premium**: Content creators in Gab who have monthly or annual subscription with Gab
- **Pro**: Users who receives benefits such as early access to features, private groups, live streaming, etc.
- **Private**: Users whose profile is accessible only to their friends

[1] https://gab.com/

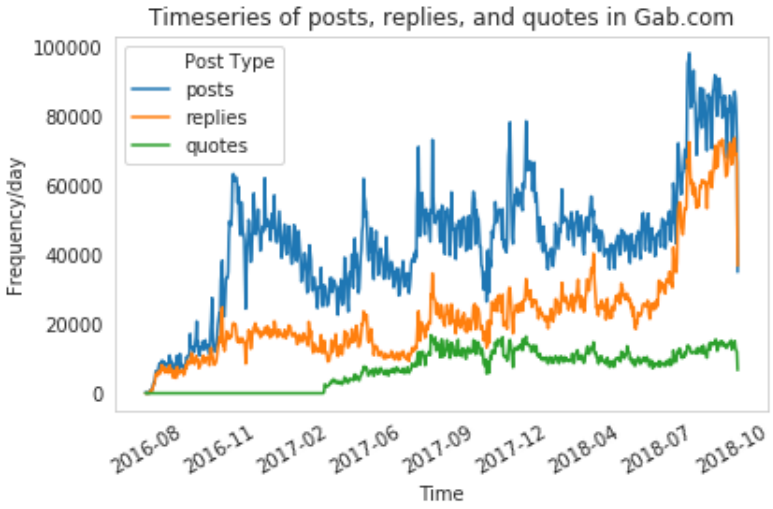

Figure 4.1: Timeseries of frequency posts, replies, and reshares from the origin of gab.com(August 2016) until the forum went down on the last week of October 2018

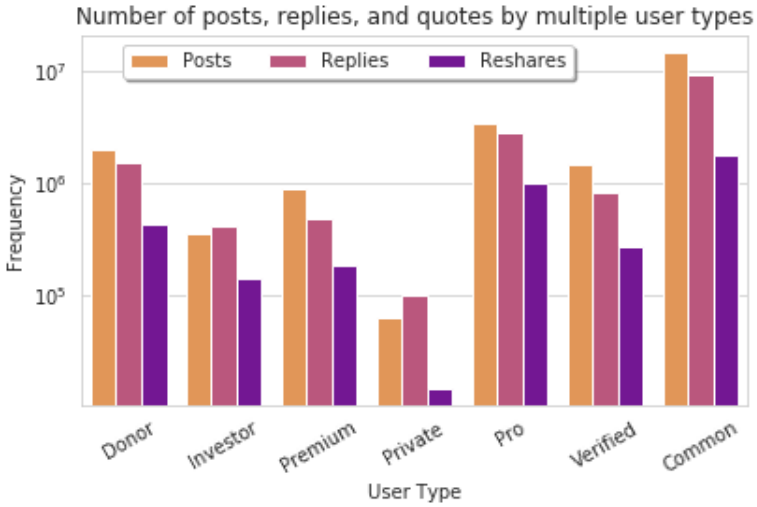

Figure 4.2: Multiple types of users in gab and their corresponding number of posts, replies, and quotes

Figure 4.3: Word cloud of hashtags mostly appeared in gab.

- **Verified**: Users who are officially verified by the Gab team to differentiate themselves from bots
- **General/Common**: Users who are neither of the above mentioned user types

Many research have showcased the presence of alt-right based conversations in the Gab media. To validate such claims, we take most frequently occurring hashtags in the posts and given them as a wordcloud in Figure 4.3. We can infer from these hashtags that users in Gab are more inclined towards US politics and politicians, and freedom of speech in the online forum. Apart from collecting all textual data, we also collected social network(*friends* and *followers*) of all public users in Gab.com.

4.2 Information Mutation In Gab

In this section, we present our framework for mining the interaction between mainstream news and Gab during shock events. While our framework is developed in the context of GAB.ai, it is easy to generalize our system across any social media platform or even combining several social media sources. As illustrated in Figure 4.4 our framework consists of three main components: sub-event detection, story chaining, detection of information mutation. We first collect posts from social media (in our

case Gab) corresponding to the event of interest. Our data comprises of posts, URLs that were shared, comments, and conversations. In order to analyze the sequence of, we further sort the data into heuristically determined time intervals. We have heuristically identified that; small amounts of data chunks are more insightful to extract sub-events. Note that the analyst can determine the timer intervals based on his/her expertise. We define sub-events as the smaller events of any particular major event. The output of Phase 1 is various sub-events, where each sub-event is captured by a cluster of words (Please see Figure 4.4). The second module is the story-chaining phase shown in Figure 4.4. In the story chaining methodology, our goal is to build a chain using the related sub-events across various time intervals. A single story chain, as captured by the algorithm, reveals the evolution or mutation of the story as a function of time. In our case study we demonstrate how a single event has multiple evolving perspectives and alternating narratives. We have formalized our algorithm in an incremental fashion wherein each sub-event in the current time interval is compared with a sub-event in the subsequent time interval and is accordingly appended to the corresponding story chain. After story chaining, we identify the news articles in social media that act as catalyst in information mutation. We first select a story chain of our interest and for each story in that time interval we find the most relevant news article and tag it to the corresponding story. The most relevant news article is found my comparing the story with the collection of news articles shared in the corresponding time interval. We do this for all the stories in story chain that are ordered by time and obtain a discourse commonality.

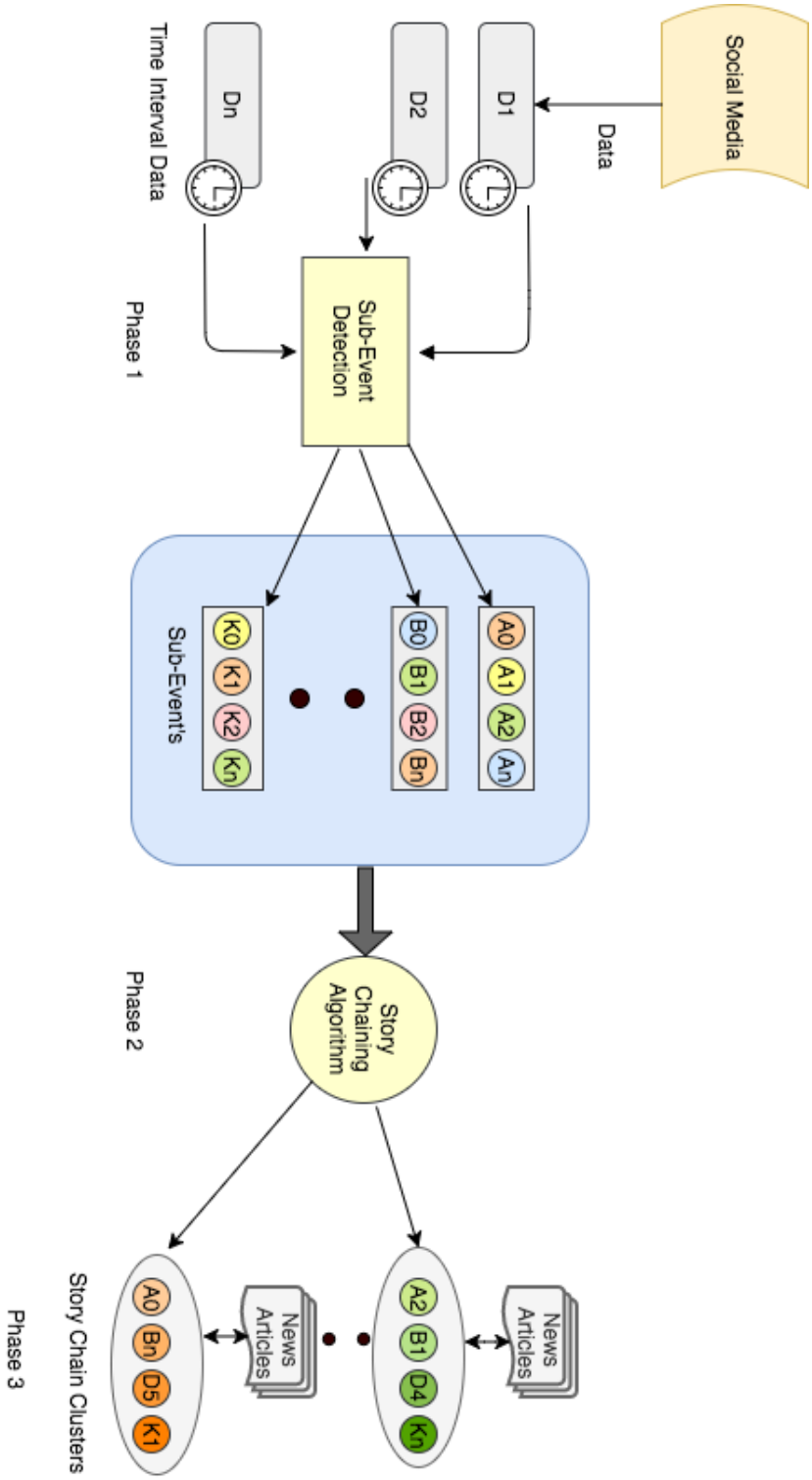

Figure 4.4: Schematic representation of our information mutation framework. Phase 1 comprises of data gathering and sub-event detection. Phase 2 takes the output of sub-event detection and chains into multiple stories using story chaining algorithm. Phase 3 of the framework finds the interaction pattern between mainstream news and social media

4.2.1 Sub-Event Detection

Most major events comprise of smaller sub-events. The first component of our pipeline captures sub-events and allows our framework to study the evolution of the captures sub-events as a function of time. We begin Phase 1 by the basic phase in text mining include preprocessing of the textual data.

4.2.1.1 Pre-Processing

Preprocessing of text produces key features or key terms from the posts made in gab.ai and enhances the relevancy between word and the clusters formed in later stages of the framework. Each post made by the user on gab.ai is represented as a feature vector which is to separate the text into individual words. This helps in selecting the significant words that carry the meaning, and remove the words that do not contribute any significance.

Punctuation do not add any value in the construction of the features. First step in pre-processing is to remove the punctuation in the text corpus. We then break-down the large chunks of text into sentences and each sentence is further divided into individual tokens also called as unigrams. The most frequently used words in a language, particularly in English,are useless in text mining. These words are called 'Stop words'. Stop-words such as pronouns, prepositions, conjunctions carry no information, for example 'the', 'of', 'and', 'to' are not important in the data.

In order to incorporate language dependent linguistic knowledge we need to find out the root of a word. Stemming converts words to their stems/root .The hypothesis of applying the stemming is that stemmed words have relative meaning in their context. We apply porter stemmer algorithm to the rest of the words after stop word removal. The final step of the preprocessing involve creation of the vector of keyword weights. We assign weights to the keywords based on the frequency of occurrence of the term in the document and the number of documents that use that term. This technique

is called TF/IDF which tells the importance of the word in a document and in the entire corpus. We create a weight matrix of all the documents and as a result we obtain vectors with the various terms along with their weight.

4.2.1.2 Algorithm

Phase 1 ingests pre-processed posts from GAB.ai that have been grouped by heuristically determined time intervals. In our observation, we found that a one-hour time window is the ideal granularity to ascertain sub-events. As described in Algorithm 1, Our sub-event detection module uses similarity measures (cosine similarity, description below) in conjunction with the single-pass centroid similarity technique **reference**. The sub-event detection module works as follows:

1. Given a set of posts, a threshold τ, we assume a set of sub-events $E_0, E_1 \dots E_m$, where m is the number of sub-events, each initialized as . Note that m can be arbitrary and we use a fixed m for explanation purposes.
2. We take each post D_i from $D_1, D_2 \dots D_n$ in an incremental fashion and compute the similarity using similarity function $F(D_i, E_j)$ = Cosine Similarity

$$F\big(D_i, E_j\big) = \frac{D_i.E_j}{||D_i||\ ||E_j||} = \frac{\sum_{l=1}^{m} D_i E_j}{\sqrt{\sum_{l=1}^{m} D_i^2}\ \sqrt{\sum_{l=1}^{m} E_j^2}}. \tag{4.1}$$

   A term frequency inverse document frequency (TF-IDF) representation is used in calculating the similarity of data point and centroid. We have considered the centroid of the sub-event E_j as the average of tf-idf score per term of the cluster.
3. We assign D_i to sub-event E if the value of $F(D_i, E_j)$ is maximum $\forall j$ and is greater than the threshold T i.e. $F(D_i, E_j) > \tau$.
4. Otherwise, we append the data point D_i to the new sub-event E_j with the centroid value as the value of D_i.

Algorithm 1 Sub-event Detection from gab.com posts

```
1: procedure FindEvents(Data features D = {D_0, D_1, ..., D_m} , Threshold τ)
2:     E = ∅
3:     for D_i ∈ D do
4:         S = ∅
5:         for E_j ∈ E do
6:             S_j = CosineSimilarity(D_i, E_j)
7:         s = max(S)
8:         if s > τ then
9:             Add D_i to E_j
10:        else
11:            Add D_i to E_{|E|}
12:    return E
```

4.2.2 Story-Chaining

The second step in analyzing the evolution or mutation of an event is to connect event clusters across multiple time intervals. Thus phase 2 of the framework utilizes the detected sub-events (output of Phase 1 - Sub-Event Detection) and chains them in an incremental fashion. Our story-chaining algorithm computes the Jaccard coefficient, described below) across time intervals to extract story-chains as given in Algorithm 2. The story-chaining algorithm works as follows:

1. Given a group of sub-events $G_0, G_1 \dots G_p$, where each G_i is a set of sub-events $E_{t_0}, E_{t_1}, \dots E_{t_m}$ and a threshold γ. We initialize a set of stories Z to
2. We incrementally take two subsequent groups G_t and $G_t + 1$ ordered by time. For each sub-event E_{t_m} in G_t we compare E_{t+1_m} of subsequent group G_{t+1} using the similarity function

$$J(E_{t_m}, E_{t+1_m}) = \frac{\text{words in } E_{t_m} \bigcap \text{words in } E_{t+1_m}}{\text{words in } E_{t_m} \bigcup \text{words in } E_{t+1_m}}. \tag{4.2}$$

3. We assign E_{t_m} and E_{t+1_m} to a story Z_k if the value of $J(E_{t_m}, E_{t+1_m})$ is maximum for all values of m and $J(E_{t_m}, E_{t+1_m}) > \gamma$. We append to Z_k if E_{t_m} exists in any Z_k.
4. Otherwise, we append E_{t_m} and E_{t+1_m} to new Z_k.

Algorithm 2 Information mutation in Gab

procedure ChainEvents(Time ordered groups of sub-events $G = \{G_0, G_1, \ldots, G_p\}$), where $G_i = \{E_{t_0}, E_{t_1}, \ldots E_{t_m}\}$ is a set of sub-events, and Threshold γ
    $\mathsf{Z} = \emptyset$
    for $G_t \in G$ **do**
        for $E_{t_m} \in G_t$ **do**
            $J = \emptyset$
            for $E_{t+1_m} \in G_{t+1}$ **do**
                Add JaccardSimilarity(E_{t_m},E_{t+1_m}) to J
            $j = max(J)$
            if $j > \gamma$ **then**
                if $E_{t_m} \subset \mathsf{Z}_k$ **then**, where $k = \{1, 2, \ldots |\mathsf{Z}|\}$
                    Add E_{t_m} to Z_k
                else
                    Add $\{E_{t_m}, E_{t+1_m}\}$ to Z_k
    return Z

4.2.3 Interaction Mining

The last phase of the framework discovers the interaction between mainstream news and story-chains of the social media posts. Phase 3 is a two-step process for each story-chain of the phase 2. In the first step, we find all the news articles posted on the Gab and group the articles by the time intervals determined in phase 1. News articles are summarized into shorter text using available text summarizer tools. In the second step, for each story in the story-chain we find the most similar news article from the group formed in step 1. The similarity between the news article and the story is computed using Jaccard similarity as given in Equation 4.2.

CHAPTER 5: RESULTS

5.0.1 Case Study 1: Charlottesville Protest

Detected subevents for the day of October 27th were analyzed for evolving narratives. A clear narrative emerged of how discussions around the breaking news evolved across the day. Within the first hour when the news broke on Gab links were posted to news about the shooting. 9/28 of these posts contained hate speech. The first alternative narrative begins to appear two hours after the shooting news broke, as the shooting being planned event in a larger conspiracy by President Donald Trumpâs political opponents to gain political power before the November 6th election. Example: https://www.puppetstringnews.com/blog/8-dead-3-cops-shot-at-pittsburgh-jewish-synagogue-shooter-yelled-all-jews-must-die (puppetstring news has been shared 15,313 on gab) During the third hour the conversation turned to discussions around false flag operations and the alt-right conspiracy theory known as âQanonâ. An alt-right support march known as â#WalkAwayâ was happening during the afternoon and news of this rally was shadowed by the release of news regarding the shooting event. Providing a motive to the conspiracy that the shooting was planned. The narratives around the shooting even begin to coalesce around the four and fifth hour with the publication of infowarâs live stream painting the shooting as âthe latest move by the Deep State to sow civil unrest and effect the historic upcoming midterm electionâ: https://www.infowars.com/breaking-alex-jones-goes-live-to-respond-to-terrorist-attack-on-pittsburgh-synagogue-the-deep-state-has-played-its-terror-card/ (Link was posted at 1:30central time and was shared 36 times during the day)

Five hours after the shooting the form of the media shared changes to become more long form, youtube videos of analysis and commentary. Blog articles explaining

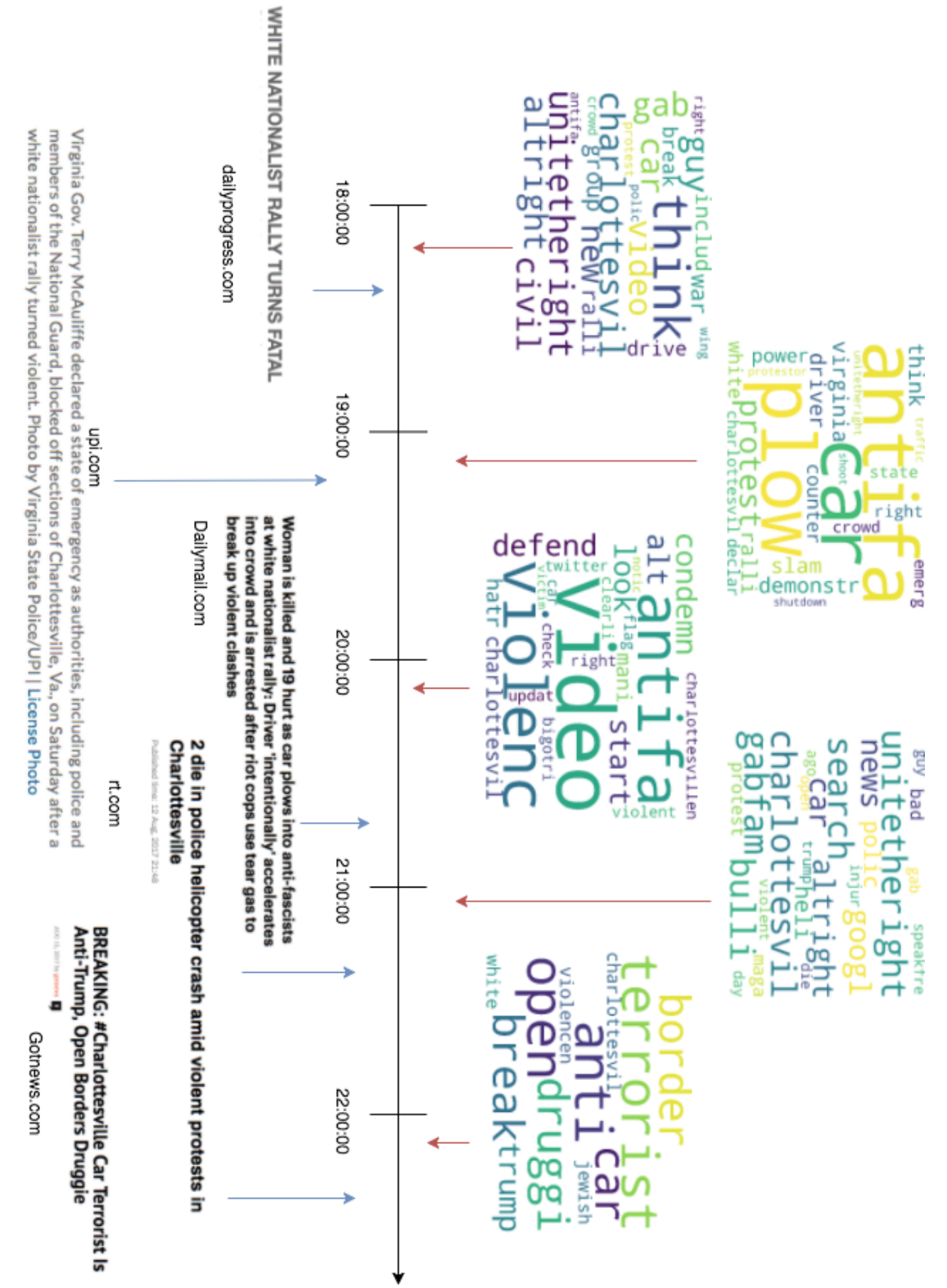

Figure 5.1: Events timeline and online discussions in Gab in effect of mainstream news articles during the Charlottesville protest. Showing the relationship of changes in topic frequency of narratives and the breaking of new news articles.

a range of conspiracies involving a deep state plot to attack the second amendment or to try to limit free speech, or to try to embarrass President trump. Our analysis shows the collective sense making [77] as people begin to provide explanations that align with their sense of reality in order to reduce anxiety and provide a sense of understanding and control of the news [78, 79].

The sub-events that were found in our analysis were analyzed for information diffusion and evolving narratives. This day are marked by narratives of blame and accusation as the support for the alt-right protesters moved from Emancipation Park to the online spaces of Gab.ai. For the day until around 2pm the discussions were consistently painting the anti-fascist counter protesters as violent while the alt-right demonstrators were peaceful and respectful. During the next hour between 2pm-3pm the first news articles begin to spread about a car driving into a crowd of protesters. Discussions are primarily based on physical observations as news consisting of images and videos is shared on the platform is focused mostly around sharing basic facts around what happened. During the second hour after the attack the discussions move from talking about the physical events into comments about the meaning of the attack and who is to blame for it. Quickly users begin to stress the narrative the domestic terrorist was not a trump supporter in attempts to distance the alt-right movement and their leader from any negative attention. Later in the hour Trump appears on national TV and blames âmany sidesâ for the violence. During the fourth hour (4pm -5pm) users revisit physical events of the day and construct evidence for various conspiracy theories. One theory that begins to take hold is that the driverâs airbag did not deploy and it must have been premeditated. Or that police had the wrong perpetrator of the attack in custody. During the fifth hour (5pm-6pm) false narratives continue to spread as the community processes new information about the attack. Rumors about the real perpetrator of the attack result in the doxing of an unrelated person who was found to have a picture of a similar looking dodge challenger

on their Facebook profile. Worries spread about the possible use of the hate crime as justification to silence the alt-rightâs ability to speak without censorship. News of a helicopter crash begin to spread and is immediately adopted into conspiracy theories connecting the two events as part of a larger organized plan. Throughout the rest of the day doxing continues of a falsely accused âopen borders druggieâ as being the perpetrator of the attack. And the prevailing narrative expands into a larger discussion around the importance of free speech and how the anti-fascist counter-protestors are not receiving blame they deserve from the mainstream media.

5.0.2 Case study 2: Pittsburgh Synagauge Shooting

Detected subevents for the day of October 27th were analyzed for evolving narratives. A clear narrative emerged on how discussions around the breaking news evolve across the day. Within the first hour when the news broke on Gab links were posted to news about the shooting. 9/28 of these posts contained hate speech. The first alternative narrative begins to appear two hours after the shooting news broke, as the shooting being planned event in a larger conspiracy by President Donald Trump‘s political opponents to gain political power before the November 6th election. Example: https://www.puppetstringnews.com/blog/8-dead-3-cops-shot-at-pittsburgh-jewish-synagogue-shooter-yelled-all-jews-must-die (puppetstring news has been shared 15,313 on gab)

During the third hour the conversation turned to discussions around false flag operations and the alt-right conspiracy theory known as "Qanon". An alt-right support march known as "#WalkAway" was happening during the afternoon and news of this rally was shadowed by the release of news regarding the shooting event. Providing a motive to the conspiracy that the shooting was planned. The narratives around the shooting even begin to coalesce around the four and fifth hour with the publication of infowarâs live stream painting the shooting as "the latest move by the Deep State to sow civil unrest and effect the historic upcoming midterm election":

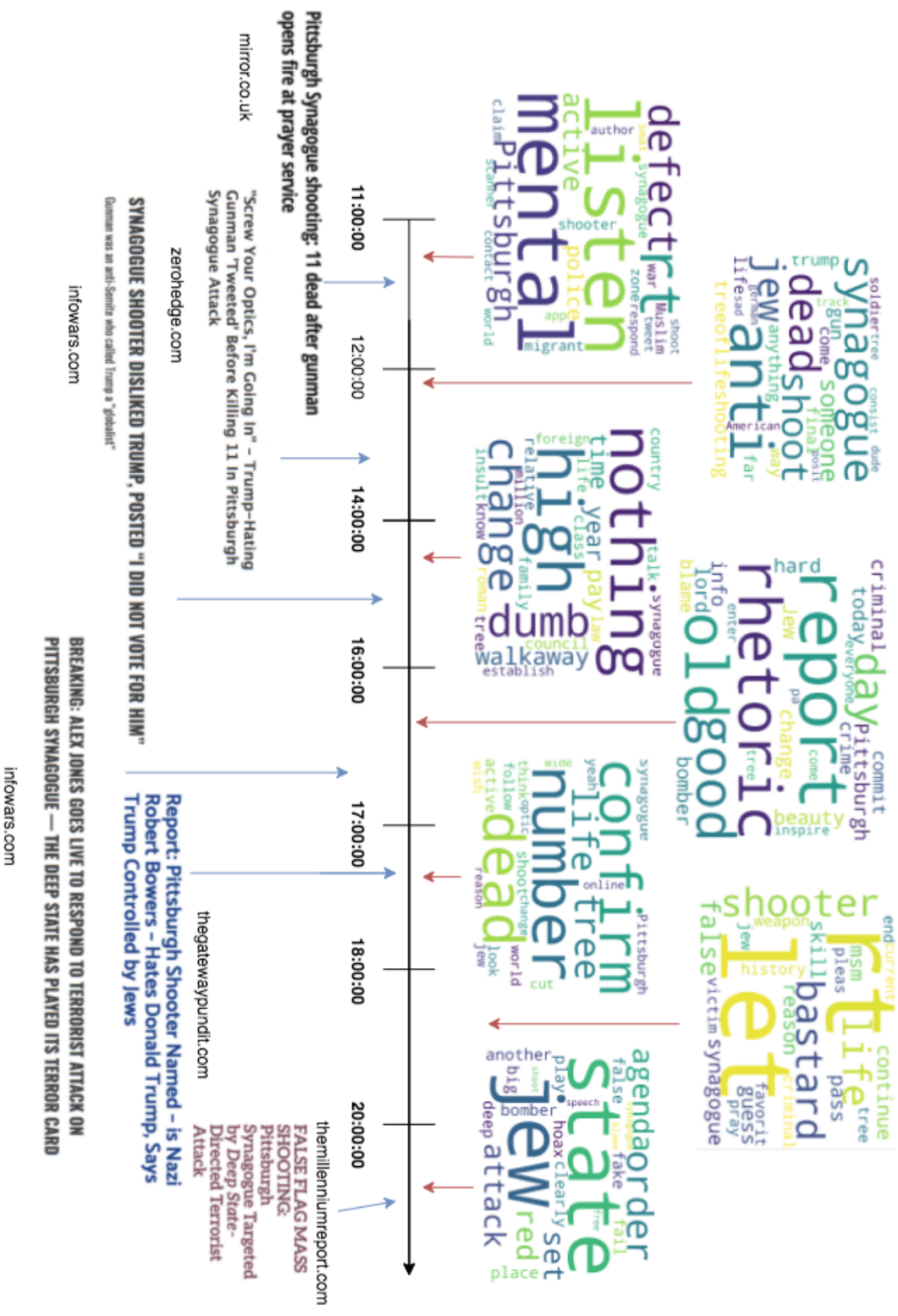

Figure 5.2: Events timeline and online discussions in Gab in effect of mainstream news articles during the Pittsburgh synagauge shooting. The diagram depicts the relationship between the stories formed through the framework and the corresponding news stories posted in the traditional news media.

https://www.infowars.com/breaking-alex-jones-goes-live-to-respond-to-terrorist-attack-on-pittsburgh-synagogue-the-deep-state-has-played-its-terror-card/ (Link was posted at 1:30central time and was shared 36 times during the day)

Five hours after the shooting the form of the media shared changes to become more long form, youtube videos of analysis and commentary. Blog articles explaining a range of conspiracies involving a deepstate plot to attack the second amendment or to try to limit free speech, or to try to embarrass President trump. Our analysis shows the collective sense making [77] as people begin to provide explanations that align with their sense of reality in order to reduce anxiety and provide a sense of understanding and control of the news [78, 79].

5.0.3 Analysis

In addition to detailed discussion on the two event case studies we further analyze some interesting results. We first started our analysis by checking the activity of users on the day of charlottesville event. As seen in Figure 5.6 we find a spike in activity of users from afternoon till post evening where there were several sub-events such as car ramming into the crowd and helicopter crash.

We further analyzed the news articles shared on the day of event on gab.ai. In our observation as shown in Figure 5.5, the number of news articles of alt-right media shared is more than twice of the main stream news media. The plot also depicts the count of the links of social media such as twitter, facebook, gab.ai, youtube and bit.ly (labeled them as others) shared on the event day. The domain cloud 5.4 depicts different news channels which are shared on gab.ai on the day of event. The size of the word in the domain cloud diagram represents the frequency of the news agency being shared.

We have plotted the effectiveness of sub-event detection as shown in the Figure 5.3. We can see the number of posts relevant to the clusters(sub-events) formed is more than the number of posts that are not relevant. We also infer that the users are

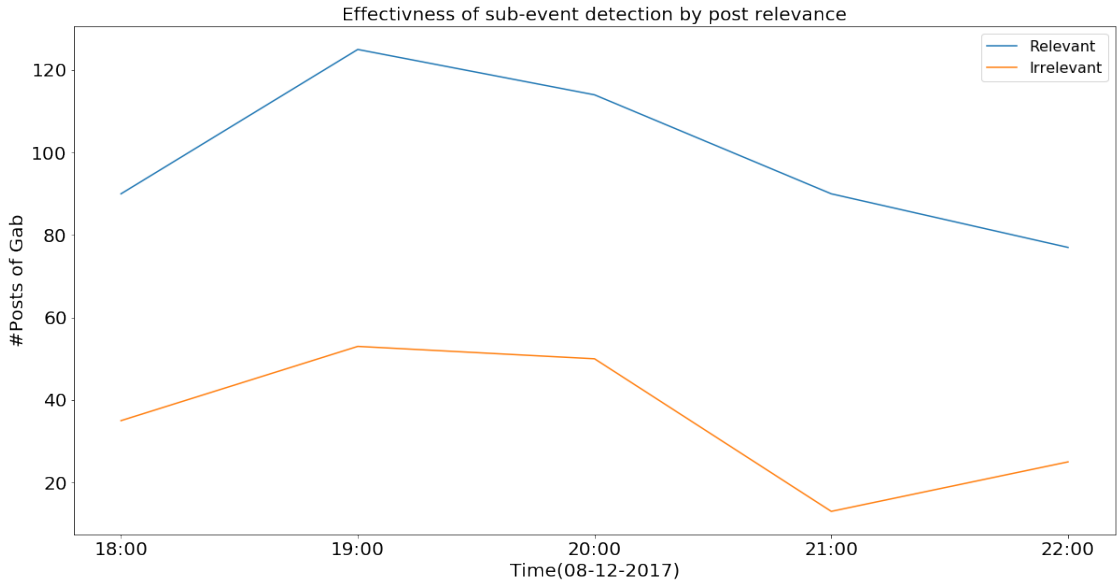

Figure 5.3: Number of posts which are relevant and irrelevant to the sub-event clusters formed in a given time. X-axis represents the timestamp by hour and Y-axis represents the number of posts made by gab.ai user per hour.

talking more about the sub-events occurred on the day of unite the rally event. As discussed in the methodology chapter the story chains are formed by comparing the clusters(sub-events) formed through our sub-event detection algorithm. The figure 5.7 represented in the form of heatmap shows the results of the jaccard similarities between sub-events of charlottesville event.

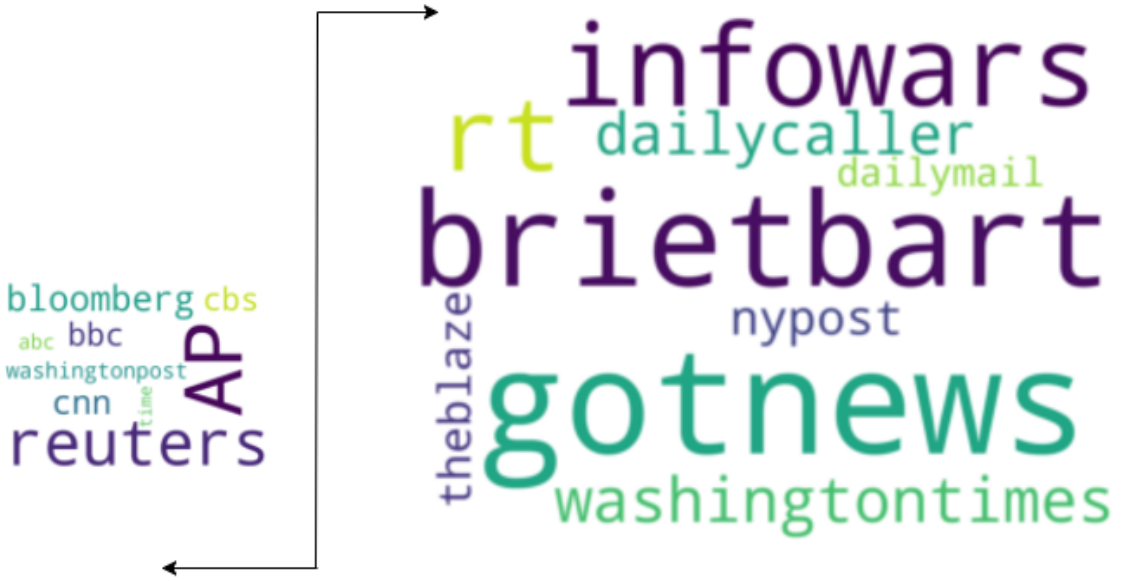

Figure 5.4: Depicts domain cloud of main-stream on left and alternative news media on right.

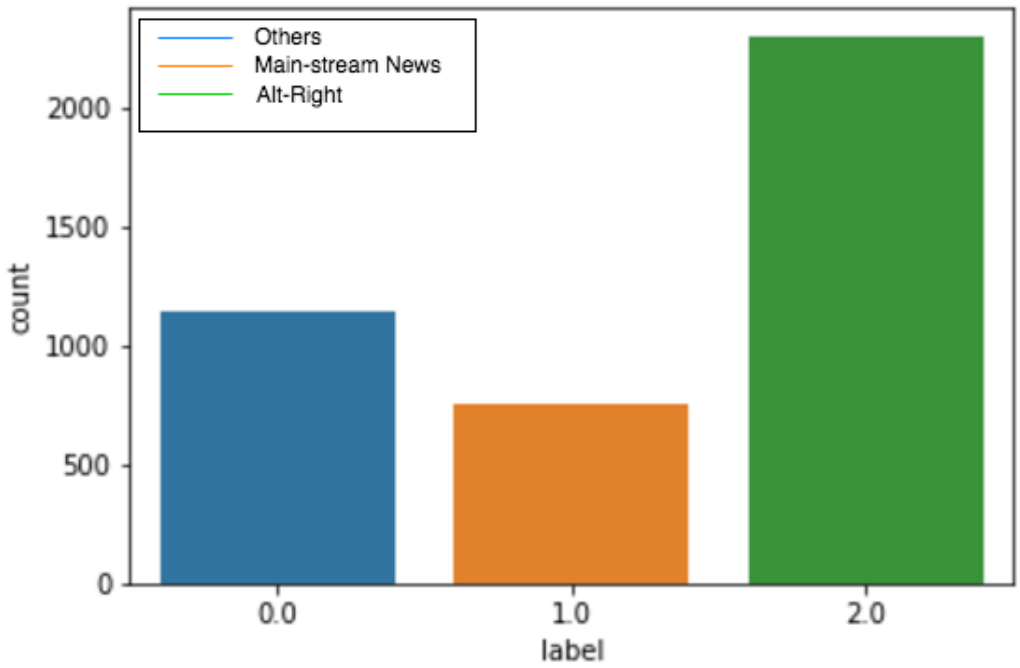

Figure 5.5: Statistics on number of news articles shared on 08/12/2017(Unite the rally event). X-axis label represents the type of news. 0-Others, 1-main stream, 2-alt right news. Y-axis represents number of such articles.

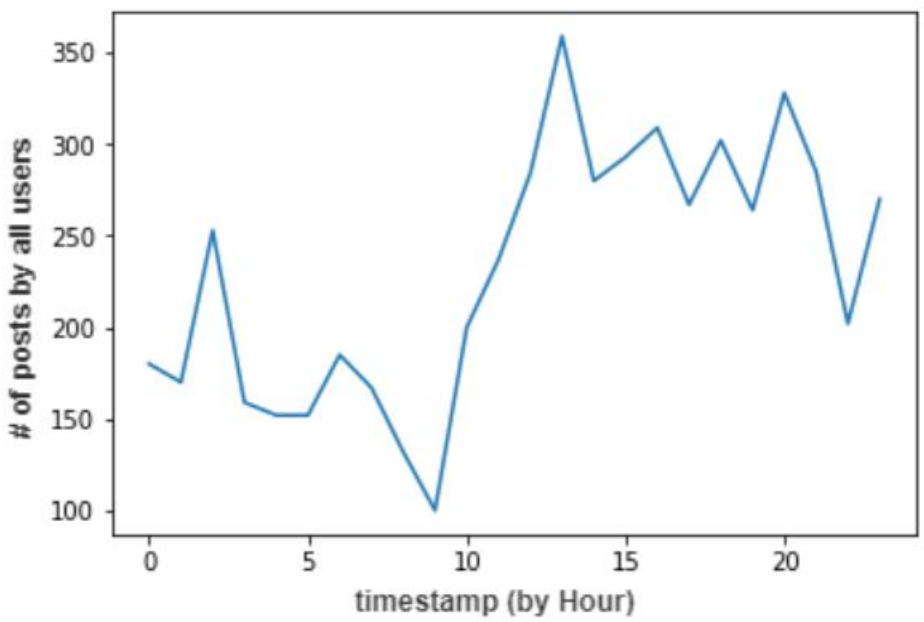

Figure 5.6: X-label represents the time by hour on 08/12/2018 and Y-axis label represents number of posts(Activity) made by all users.

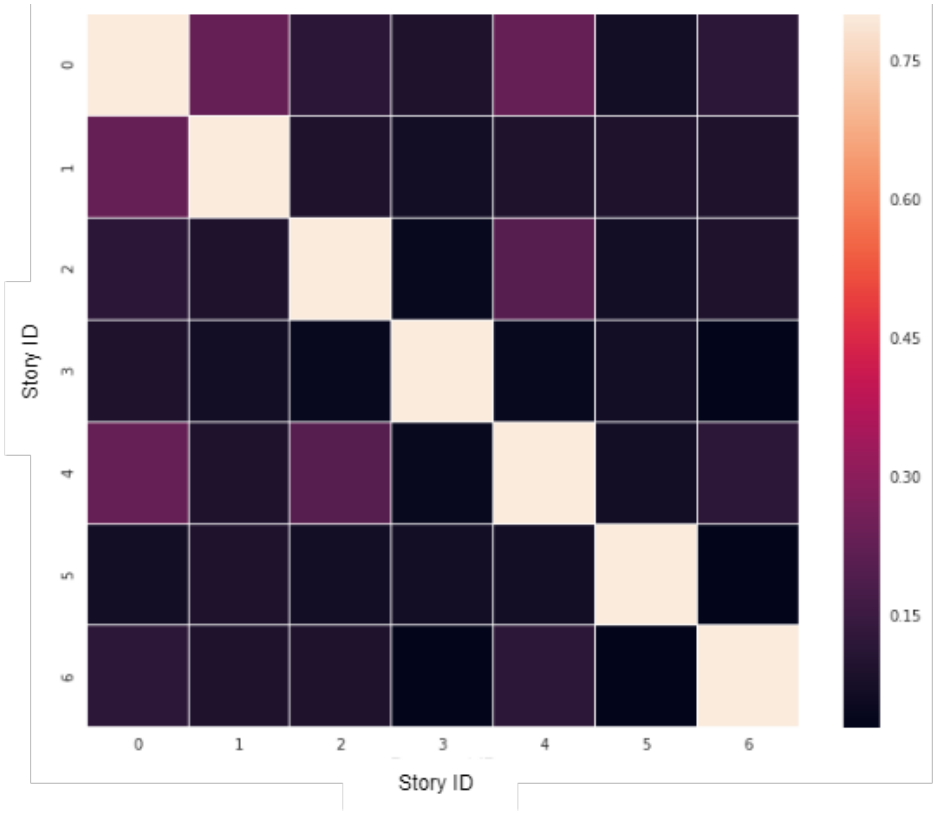

Figure 5.7: Heat map of Jaccard similarities of the Sub-events in a story chain. X-axis and Y-axis represents stories formed by Sub-Event detection algorithm in the form of clusters. Each story ID is a cluster of words.

CHAPTER 6: CONCLUSIONS

In this work, we presented a framework to examine information mutation in a social system like Gab during the outbreak of events. We could able to characterize how there is a disagreement between the reporting of major news sites and the opinions expressed in communities like Gab. We tested the proposed framework for two events such as Unite the Right Charlottesville protests and Pittsburgh Tree of Life Synagogue shooting. We presented detailed analysis on the activity of users and the domains shared on the days of events. We showed that how rapidly and frequently alternative news information was shared on Gab.

Our framework is not only restricted to gab.ai textual data but is flexible to take any other social media data including temporal, geographical and other actors as the features. We have restricted the number of words formed in each cluster of sub-event detection to a fixed size in order to have consistency among the stories but the framework allows to vary the size of the these clusters.

For future work, we would like to apply our framework using the data of other social media such as twitter, reddit etc. Since our framework uses only the textual data as a feature, we would like to merge the textual data from different sources and feed on our framework. Another interesting direction is to not only use textual features but also temporal features of Gab. We also plan to extend our framework in forecasting the entities in the stories formed through our algorithm.

REFERENCES

[1] S. Zannettou, B. Bradlyn, E. De Cristofaro, M. Sirivianos, G. Stringhini, H. Kwak, and J. Blackburn, "What is gab? a bastion of free speech or an alt-right echo chamber?," *arXiv preprint arXiv:1802.05287*, 2018.

[2] L. Lima, J. C. Reis, P. Melo, F. Murai, L. Araujo, P. Vikatos, and F. Benevenuto, "Inside the right-leaning echo chambers: Characterizing gab, an unmoderated social system," in *2018 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*, pp. 515–522, IEEE, 2018.

[3] K. Starbird, "Examining the alternative media ecosystem through the production of alternative narratives of mass shooting events on twitter.," in *ICWSM*, pp. 230–239, 2017.

[4] S. González-Bailón, J. Borge-Holthoefer, A. Rivero, and Y. Moreno, "The dynamics of protest recruitment through an online network," *CoRR*, vol. abs/1111.5595, 2011.

[5] C. C. Aggarwal and C. Zhai, *A Survey of Text Clustering Algorithms*, pp. 77–128. Boston, MA: Springer US, 2012.

[6] J. Allan, *Topic Detection and Tracking: Event-based Information Organization*. Springer Publishing Company, Incorporated, 2012.

[7] Y. Yang, T. Pierce, and J. Carbonell, "A study of retrospective and on-line event detection," in *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '98, (New York, NY, USA), pp. 28–36, ACM, 1998.

[8] Y. Yang, J. Zhang, J. Carbonell, and C. Jin, "Topic-conditioned novelty detection," in *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '02, (New York, NY, USA), pp. 688–693, ACM, 2002.

[9] G. Salton, *Automatic Text Processing: The Transformation, Analysis, and Retrieval of Information by Computer*. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 1989.

[10] G. Kumaran and J. Allan, "Text classification and named entities for new event detection," in *Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '04, (New York, NY, USA), pp. 297–304, ACM, 2004.

[11] M. Mohd, "Named entity patterns across news domains,"

[12] Y. Yang, J. G. Carbonell, R. D. Brown, T. Pierce, B. T. Archibald, and X. Liu, "Learning approaches for detecting and tracking news events," *IEEE Intelligent Systems*, vol. 14, pp. 32–43, July 1999.

[13] Z. Li, B. Wang, M. Li, and W.-Y. Ma, "A probabilistic model for retrospective news event detection," in *Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '05, (New York, NY, USA), pp. 106–113, ACM, 2005.

[14] T. Leek, R. Schwartz, and S. Sista, "Topic detection and tracking," ch. Probabilistic Approaches to Topic Detection and Tracking, pp. 67–83, Norwell, MA, USA: Kluwer Academic Publishers, 2002.

[15] T. Brants, F. Chen, and A. Farahat, "A system for new event detection," in *Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Informaion Retrieval*, SIGIR '03, (New York, NY, USA), pp. 330–337, ACM, 2003.

[16] T. Jo and M. Lee, "The evaluation measure of text clustering for the variable number of clusters," in *Advances in Neural Networks – ISNN 2007* (D. Liu, S. Fei, Z. Hou, H. Zhang, and C. Sun, eds.), (Berlin, Heidelberg), pp. 871–879, Springer Berlin Heidelberg, 2007.

[17] J. Allan, R. Papka, and V. Lavrenko, "On-line new event detection and tracking," in *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '98, (New York, NY, USA), pp. 37–45, ACM, 1998.

[18] P. Berkhin, "A survey of clustering data mining techniques," *Grouping Multidimensional Data*, pp. 25–71, 2006.

[19] C. C. Aggarwal and C. Zhai, "A survey of text clustering algorithms.," in *Mining Text Data* (C. C. Aggarwal and C. Zhai, eds.), pp. 77–128, Springer, 2012.

[20] A. K. Jain and R. C. Dubes, *Algorithms for Clustering Data*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 1988.

[21] X.-Y. Dai, Q. Chen, X. Wang, and J. Xu, "Online topic detection and tracking of financial news based on hierarchical clustering," *2010 International Conference on Machine Learning and Cybernetics*, vol. 6, pp. 3341–3346, 2010.

[22] C. Bouras and V. Tsogkas, "Assigning web news to clusters," in *Proceedings of the 2010 Fifth International Conference on Internet and Web Applications and Services*, ICIW '10, (Washington, DC, USA), pp. 1–6, IEEE Computer Society, 2010.

[23] G. Luo, C. Tang, and P. S. Yu, "Resource-adaptive real-time new event detection," in *Proceedings of the 2007 ACM SIGMOD International Conference on Management of Data*, SIGMOD '07, (New York, NY, USA), pp. 497–508, ACM, 2007.

[24] R. Papka, *On-line New Event Detection, Clustering, and Tracking (Information Retrieval, Internet)*. PhD thesis, 1999. AAI9950198.

[25] H. Becker, M. Naaman, and L. Gravano, "Selecting quality twitter content for events," 2011.

[26] C. Castillo, M. Mendoza, and B. Poblete, "Information credibility on twitter," in *Proceedings of the 20th International Conference on World Wide Web*, WWW '11, (New York, NY, USA), pp. 675–684, ACM, 2011.

[27] J. Hurlock and M. L. Wilson, "Searching twitter: Separating the tweet from the chaff.," in *ICWSM* (L. A. Adamic, R. A. Baeza-Yates, and S. Counts, eds.), The AAAI Press, 2011.

[28] K. Lee, B. D. Eoff, and J. Caverlee, "Seven months with the devils: A long-term study of content polluters on twitter.," in *ICWSM* (L. A. Adamic, R. A. Baeza-Yates, and S. Counts, eds.), The AAAI Press, 2011.

[29] A. Kontostathis, L. M. Galitsky, W. M. Pottenger, S. Roy, and D. J. Phelps, "A survey of emerging trend detection in textual data mining," in *Survey of Text Mining*, pp. 185–224, Springer, 2004.

[30] J. Kleinberg, "Bursty and hierarchical structure in streams," in *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '02, (New York, NY, USA), pp. 91–101, ACM, 2002.

[31] G. P. C. Fung, J. X. Yu, P. S. Yu, and H. Lu, "Parameter free bursty events detection in text streams," in *Proceedings of the 31st International Conference on Very Large Data Bases*, VLDB '05, pp. 181–192, VLDB Endowment, 2005.

[32] Q. He, K. Chang, and E.-P. Lim, "Analyzing feature trajectories for event detection," in *Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '07, (New York, NY, USA), pp. 207–214, ACM, 2007.

[33] Q. He, K. Chang, E. peng Lim, and J. Zhang, "Bursty feature representation for clustering text streams."

[34] X. Wang, C. Zhai, X. Hu, and R. Sproat, "Mining correlated bursty topic patterns from coordinated text streams," in *Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '07, (New York, NY, USA), pp. 784–793, ACM, 2007.

[35] S. Goorha and L. Ungar, "Discovery of significant emerging trends," in *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '10, (New York, NY, USA), pp. 57–64, ACM, 2010.

[36] T. Snowsill, F. Nicart, M. Stefani, T. D. Bie, and N. Cristianini, "Finding surprising patterns in textual data streams," *2010 2nd International Workshop on Cognitive Information Processing*, pp. 405–410, 2010.

[37] M. Mathioudakis and N. Koudas, "Twittermonitor: Trend detection over the twitter stream," in *Proceedings of the 2010 ACM SIGMOD International Conference on Management of Data*, SIGMOD '10, (New York, NY, USA), pp. 1155–1158, ACM, 2010.

[38] M. Naaman, H. Becker, and L. Gravano, "Hip and trendy: Characterizing emerging trends on twitter," *J. Am. Soc. Inf. Sci. Technol.*, vol. 62, pp. 902–918, May 2011.

[39] J. Sankaranarayanan, H. Samet, B. E. Teitler, M. D. Lieberman, and J. Sperling, "Twitterstand: News in tweets," in *Proceedings of the 17th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, GIS '09, (New York, NY, USA), pp. 42–51, ACM, 2009.

[40] S. Phuvipadawat and T. Murata, "Breaking news detection and tracking in twitter," in *Proceedings of the 2010 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology - Volume 03*, WI-IAT '10, (Washington, DC, USA), pp. 120–123, IEEE Computer Society, 2010.

[41] S. Petrović, M. Osborne, and V. Lavrenko, "Streaming first story detection with application to twitter," in *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, HLT '10, (Stroudsburg, PA, USA), pp. 181–189, Association for Computational Linguistics, 2010.

[42] J. Allan, V. Lavrenko, and H. Jin, "First story detection in tdt is hard," in *Proceedings of the Ninth International Conference on Information and Knowledge Management*, CIKM '00, (New York, NY, USA), pp. 374–381, ACM, 2000.

[43] A. Gionis, P. Indyk, and R. Motwani, "Similarity search in high dimensions via hashing," in *Proceedings of the 25th International Conference on Very Large Data Bases*, VLDB '99, (San Francisco, CA, USA), pp. 518–529, Morgan Kaufmann Publishers Inc., 1999.

[44] R. Long, H. Wang, Y. Chen, O. Jin, and Y. Yu, "Towards effective event detection, tracking and summarization on microblog data," in *Proceedings of the 12th International Conference on Web-age Information Management*, WAIM'11, (Berlin, Heidelberg), pp. 652–663, Springer-Verlag, 2011.

[45] M. Cordeiro, "Twitter event detection: Combining wavelet analysis and topic inference summarization," in *Doctoral Symposium on Informatics Engineering, DSIE*, 2012.

[46] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent dirichlet allocation," *J. Mach. Learn. Res.*, vol. 3, pp. 993–1022, Mar. 2003.

[47] A.-M. Popescu and M. Pennacchiotti, "Detecting controversial events from twitter," CIKM '10, (New York, NY, USA), pp. 1873–1876, ACM, 2010.

[48] J. H. Friedman, "Greedy function approximation: A gradient boosting machine," *Annals of Statistics*, vol. 29, pp. 1189–1232, 2000.

[49] A.-M. Popescu, M. Pennacchiotti, and D. Paranjpe, "Extracting events and event descriptions from twitter," in *Proceedings of the 20th International Conference Companion on World Wide Web*, WWW '11, (New York, NY, USA), pp. 105–106, ACM, 2011.

[50] D. Paranjpe, "Learning document aboutness from implicit user feedback and document structure," in *Proceedings of the 18th ACM Conference on Information and Knowledge Management*, CIKM '09, (New York, NY, USA), pp. 365–374, ACM, 2009.

[51] E. Benson, A. Haghighi, and R. Barzilay, "Event discovery in social media feeds," in *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1*, HLT '11, (Stroudsburg, PA, USA), pp. 389–398, Association for Computational Linguistics, 2011.

[52] R. Lee and K. Sumiya, "Measuring geographical regularities of crowd behaviors for twitter-based geo-social event detection," in *Proceedings of the 2Nd ACM SIGSPATIAL International Workshop on Location Based Social Networks*, LBSN '10, (New York, NY, USA), pp. 1–10, ACM, 2010.

[53] T. Fujisaka, R. Lee, and K. Sumiya, "Discovery of user behavior patterns from geo-tagged micro-blogs," in *Proceedings of the 4th International Conference on Uniquitous Information Management and Communication*, ICUIMC '10, (New York, NY, USA), pp. 36:1–36:10, ACM, 2010.

[54] T. Sakaki, M. Okazaki, and Y. Matsuo, "Earthquake shakes twitter users: Real-time event detection by social sensors," in *Proceedings of the 19th International Conference on World Wide Web*, WWW '10, (New York, NY, USA), pp. 851–860, ACM, 2010.

[55] D. Fox, J. Hightower, L. Liao, D. Schulz, and G. Borriello, "Bayesian filtering for location estimation," *IEEE Pervasive Computing*, vol. 2, pp. 24–33, July 2003.

[56] K. Massoudi, M. Tsagkias, M. de Rijke, and W. Weerkamp, "Incorporating query expansion and quality indicators in searching microblog posts," in *Proceedings of the 33rd European Conference on Advances in Information Retrieval*, ECIR'11, (Berlin, Heidelberg), pp. 362–367, Springer-Verlag, 2011.

[57] W. Weerkamp and M. de Rijke, "Credibility improves topical blog post retrieval," in *Proceedings of ACL-08: HLT*, pp. 923–931, Association for Computational Linguistics, 2008.

[58] A. Campan, A. Cuzzocrea, and T. M. Truta, "Fighting fake news spread in online social networks: Actual trends and future research directions," in *2017 IEEE International Conference on Big Data (Big Data)*, pp. 4453–4457, Dec 2017.

[59] R. Ottoni, E. Cunha, G. Magno, P. Bernadina, W. Meira Jr, and V. Almeida, "Analyzing right-wing youtube channels: Hate, violence and discrimination," *in Proceedings of the 10th ACM Conference on Web Science*, 2018.

[60] K. Starbird, A. Arif, T. Wilson, K. V. Koevering, K. Yefimova, and D. Scarnecchia, "Ecosystem or echo-system? exploring content sharing across alternative media domains," in *Proceedings of the Twelfth International Conference on Web and Social Media, ICWSM 2018, Stanford, California, USA, June 25-28, 2018.*, pp. 365–374, 2018.

[61] A. Arif, L. G. Stewart, and K. Starbird, "Acting the part: Examining information operations within# blacklivesmatter discourse," *Proceedings of the ACM on Human-Computer Interaction*, vol. 2, no. CSCW, p. 20, 2018.

[62] S. Zannettou, T. Caulfield, E. De Cristofaro, N. Kourtelris, I. Leontiadis, M. Sirivianos, G. Stringhini, and J. Blackburn, "The web centipede: understanding how web communities influence each other through the lens of mainstream and alternative news sources," in *Proceedings of the 2017 Internet Measurement Conference*, pp. 405–417, ACM, 2017.

[63] L. A. Silva, M. Mondal, D. Correa, F. Benevenuto, and I. Weber, "Analyzing the targets of hate in online social media.," in *ICWSM*, pp. 687–690, 2016.

[64] M. Mondal, L. A. Silva, and F. Benevenuto, "A measurement study of hate speech in social media," in *Proceedings of the 28th ACM Conference on Hypertext and Social Media*, pp. 85–94, ACM, 2017.

[65] M. C. Benigni, "Detection and analysis of online extremist communities," 2017.

[66] E. Ferrara, W.-Q. Wang, O. Varol, A. Flammini, and A. Galstyan, "Predicting online extremism, content adopters, and interaction reciprocity," in *International conference on social informatics*, pp. 22–39, Springer, 2016.

[67] Y. Ning, S. Muthiah, R. Tandon, and N. Ramakrishnan, "Uncovering news-twitter reciprocity via interaction patterns," in *Proceedings of the 2015 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining 2015*, pp. 1–8, ACM, 2015.

[68] L. Hong, W. Yang, P. Resnik, and V. Frias-Martinez, "Uncovering topic dynamics of social media and news: The case of ferguson," in *Social Informatics* (E. Spiro and Y.-Y. Ahn, eds.), (Cham), pp. 240–256, Springer International Publishing, 2016.

[69] K. Lerman and R. Ghosh, "Information contagion: an empirical study of the spread of news on digg and twitter social networks," *CoRR*, vol. abs/1003.2664, 2010.

[70] R. C. Barranco, A. P. Boedihardjo, and M. S. Hossain, "Analyzing evolving stories in news articles," *CoRR*, vol. abs/1703.08593, 2017.

[71] H. MS, G. J, E. Y, H. R, P. M, and R. N, "Connecting the dots between pubmed abstracts," 2012.

[72] J. Schlachter, A. Ruvinsky, L. A. Reynoso, S. Muthiah, and N. Ramakrishnan, "Leveraging topic models to develop metrics for evaluating the quality of narrative threads extracted from news stories," 2015.

[73] D. Shahaf and C. Guestrin, "Connecting the dots between news articles," in *Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence - Volume Volume Three*, IJCAI'11, pp. 2734–2739, AAAI Press, 2011.

[74] R. C. Barranco, A. P. Boedihardjo, and M. S. Hossain, "Analyzing evolving stories in news articles," *International Journal of Data Science and Analytics*, pp. 1–16, 2015.

[75] L. Hong, W. Yang, P. Resnik, and V. Frias-Martinez, "Uncovering topic dynamics of social media and news: the case of ferguson," in *International Conference on Social Informatics*, pp. 240–256, Springer, 2016.

[76] D. Abhik and D. Toshniwal, "Sub-event detection during natural hazards using features of social media data," in *Proceedings of the 22nd International Conference on World Wide Web*, pp. 783–788, ACM, 2013.

[77] R. Meyersohn, "Improvised news: A sociological study of rumor," 1969.

[78] J.-W. van Prooijen and M. Acker, "The influence of control on belief in conspiracy theories: Conceptual and applied extensions," *Applied Cognitive Psychology*, vol. 29, no. 5, pp. 753–761, 2015.

[79] P. Mandik, "Shit happens," *Episteme*, vol. 4, no. 2, pp. 205–218, 2007.