[1,2] **Ruhi Patankar,**

[1] **A.Pravin,**

[2] **Shreya Mukherjee**

Twitter Event Tracking Based on Segmentation

Abstract: - The domain of text mining has witnessed an increase in interest regarding event detection, especially with the wealth of information accessible on social media sites such as Twitter. Twitter's unique features such as hashtags and character limits enable quick reporting of real-world events, making it an invaluable resource. While previous studies have mainly focused on localized or breaking news events, many significant occurrences have been overlooked. This paper tackles the challenges of event identification using Twitter and presents SEDTWik, a system that leverages tweet segmentation to identify noteworthy events across various locations and categories. The approach involves segmenting tweets and hashtags, detecting bursty segments, clustering them, and summarizing the results. Evaluation on the Events2012 corpus demonstrates the system's outstanding performance. Key terms include Wikipedia, text mining, Twitter, social media, microblogging, tweet segmentation, and event detection.

Keywords: Text Mining, Event Detection, Segmentation, Clustering, Bursty Segments, Twitter

I. INTRODUCTION

Microblogging has become increasingly prominent in recent years, with platforms like Twitter leading the way with its 280-character limit per tweet. Twitter is a platform for sharing happenings in real-time as well as a way to communicate with others. Events are distinct, as defined by initiatives such as Topic Detection and Tracking (TDT) and Becker et al. occurrences happening within a specific time frame and accompanied by a stream of tweets discussing the event. Users on Twitter can not only share about events but also amplify them through retweets and hashtags. Hashtags, such as #RIP, can signify the nature of the event, while some may serve as vehicles for promoting memes or ideas. However, detecting events from tweets faces challenges like dealing with noisy, informal language and a vast amount of data being generated at rapid rates. We present SEDTWik, an event detection system based on tweet segmentation that uses external information sources such as Wikipedia to overcome these difficulties. The subsequent sections of this paper outline the methodology of SEDTWik, encompassing event summarization, bursty segment extraction, and tweet segmentation. The results section follows, detailing the experimental setup, segmentation statistics, event detection findings, the impact of H and T, and a comparative analysis using the Elbow Method combined with NMF. Finally, the paper discusses related work and concludes with insights for future research.

II. METHODOLOGY

This section introduces SEDTWik, an enhanced framework for event detection that builds on the Twevent system by Li et al. (2012a). SEDTWik processes tweets within a fixed time window, *t*, to identify events through a multi-step approach consisting of four key components. First, Tweet Segmentation splits tweets into meaningful units to extract relevant entities or phrases, with emphasis on hashtags and named entities to improve segment quality. Next, Bursty Segment Extraction identifies segments exhibiting unusual activity

patterns, highlighting potential event indicators. In the Bursty Segment Clustering phase, related segments are grouped into coherent event clusters using similarity measures. Finally, Event Summarization generates concise representations of identified events using the LexRank algorithm, providing a comprehensive view of the detected events. Throughout these stages, statistical metrics such as retweet counts, follower influence, and temporal dynamics are leveraged to refine and prioritize event candidate. The following subsections provide a detailed explanation of each component and how it fits into the overall event detection process.

[1] [1]*Department of CSE, Sathyabama Institute of Science and Technology, Chennai, India,*

[2]*Department of Computer Engineering and Technology, Dr. Vishwanath Karad MIT World Peace University, Pune, India*

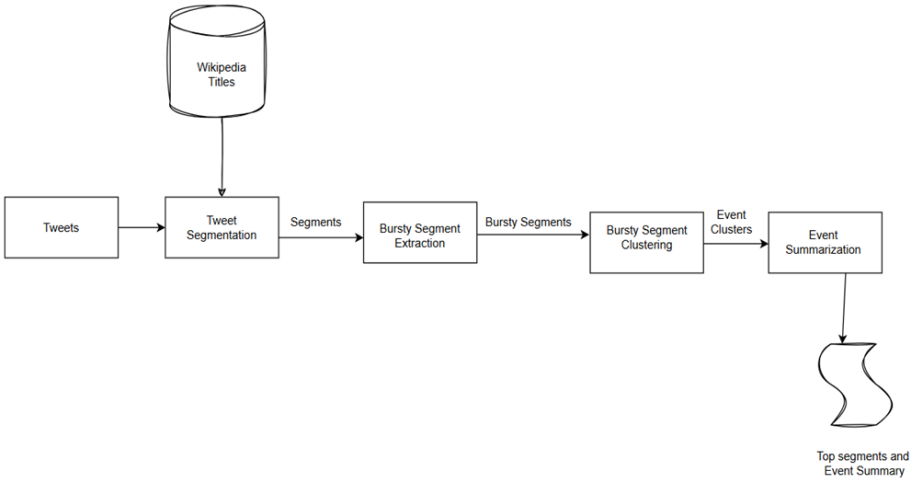

FIG.1 THE ARCHITECTURE OF SEDTWIK

A. Tweet Segmentation

In this section, we propose an alternative approach to tweet segmentation, building upon the concept introduced by Li et al. (2012b) and Li et al. (2015) for Named Entity Recognition (NER). Instead of their dynamic programming-based method, we partition tweets and hashtags using the Wikipedia Page Titles Dataset. Tweet segmentation involves splitting a tweet into meaningful, non-overlapping chunks, known as multi-grams or unigrams that can be single words or phrases. Phrases are preferred as they convey more specific information. For instance, "[vice presidential debate]" offers more context than the individual words "vice," "presidential," and "debate" separately.

During segmentation, we consider three key components: the Hashtags, name mentions, and content in tweets. Tweet text excluding URLs, hashtags, and name mentions, is filtered to retain only segments that match the titles of Wikipedia pages. This ensures that only named entities or meaningful segments are preserved, reducing noise in event detection. Name mentions—which are usually used to allude to a person's username, such as "@iamsrk" for Shah Rukh Khan—are taken as parts and are substituted with the real name of the mentioned person. Hashtags, containing condensed information, play a crucial role. Inspired by the work of Ozdikis et al. (2012a), who achieved better results by focusing solely on hashtags, we assign them greater weight in the segmentation process. We introduce a hashtag weight, denoted as (lambda), where (lambda = 2) indicates duplication of all hashtags during segmentation, effectively doubling their weight. This prioritization ensures that even if a segment isn't found in Wikipedia titles, its presence as a hashtag makes it more prominent. Furthermore, hashtags are segmented based on capitalization, with those containing no capitalization treated as unigrams. "#BreakingNews" could be divided into "[breaking news]" segments, for instance.

Overall, our approach aims to enhance event detection by effectively leveraging tweet text, name mentions, and hashtags, with a particular emphasis on the latter's weight and segmentation.

B. Bursty Segment Extraction

Once tweets are segmented, identifying abnormally bursty segments, which likely correlate with events, is crucial for efficient event detection. However, clustering all segments is computationally intensive due to the vast number of unique segments generated daily. Hence, we focus on extracting these bursty segments and discarding the rest. Let N_t be the total number of tweets in the current time window t, and let f{s,t} be the number of unique users using segment s during time window t. Retweets, defined by Boyd et al. (2010) as "a conversational practice", can indicate the significance of a tweet within a conversation. The total number of retweets for all tweets that contain segment s in t is how we get the segment retweet count, or src_{s,t}. Similarly, tweets from users with substantial follower counts can carry more weight. As a result, the segment follower count sfc_{s,t} is defined as the total of all users' follower counts for segment s in t.

The bursty weight wb(s, t) for segment s in t is found using formula (2), where p_s is the expected probability of encountering segment s in any random time window. . We use parameters to estimate the probability distribution to a Normal distribution, taking into account the huge N_t in tweets.

$$P_b(s,t)=S(10\cdot((f_{s,t}-(E[s|t]+\sigma[s|t]))/\ \sigma[s|t])) \quad (1)$$

The bursty weight determines the significance of segments, with top segments selected as bursty based on this weight. The selection of K, representing the top segments, is crucial, balancing recall and noise in event detection. A segment with f_{s,t} > E[s|t] is deemed bursty, while those with f_{s,t} < E[s|t] are discarded.

The frequency of a bursty segment is transferred to the range (0,1) by formula (1), which defines the bursty probability Pb(s,t) for segment s in the time window t.

We observe that tweets garnering numerous retweets may signify an important event, warranting greater consideration for the segments within those retweeted tweets. We quantify this significance through the segment retweet count src_s, which totals the retweet counts of all tweets containing segment s within a given time frame t. Additionally, tweets from individuals with extensive follower bases, such as celebrities or news outlets, often carry more weight and significance. To account for this, we introduce the segment follower count sfc_s, which sums the follower counts of all users employing segment s within time frame t.

$$w_b(s,t)=P_b(s,t)\cdot\log(u_{s,t)}x\log(src_{s,t})\cdot\log(\log(sfc_{s,t})) \quad (2)$$

Formula (2) defines for segment s, the bursty weight wb(s, t) inside time frame t, which is determined by integrating these measures. This weight calculation integrates both the retweet and follower counts to gauge the importance of a segment within the context of event detection. This approach ensures that tweets with substantial retweet counts and those originating from users with significant follower bases are given greater weight, thus mitigating the influence of spam or self-promotional tweets on the accuracy of event detection.

The choice of K is set to sqrt{N_t} to balance recall and noise in event detection. This process allows us to identify and prioritize bursty segments, which are likely indicative of significant events, while filtering out noise and less impactful segments.

C. Bursty Segment Clustering

In this section, we employ the clustering technique outlined by Li et al. (2012a) to group bursty segments and subsequently filter out non-event clusters. Based on the temporal frequency of two parts and the content of tweets that contain them, we calculate how similar they are, taking into account the quick and dynamic nature of tweet subjects.

We evenly split each time frame into the M sub-windows, represented as t = < t_1, t_2,..., t_M>, to accommodate for the temporal dynamics Assume that T_t(s_m) is the concatenation of every tweet that contains segment s in sub-window t_m, and let f_t(s_m) be the tweet frequency of segment s in sub-window t_m.

$$simt(sa,sb)=\Sigma wt(sa,m)wt(sb,m)\ x\ sim(Tt(sa,m),Tt(sb,m))\ \ m=1\ to\ M \quad (3)$$

Using Formula (3), the similarity sim_t(s_a, s_b) was computed. Let f_t(s_m) be the segment s tweet frequency in sub-window t_m, and assume that T_t(s_m) is the concatenation of all tweets that contain segment s in sub-window t_m. where sim(T1 T2) is the tf-idf similarity of the set of tweets T1 and T2, and wt(sm) is the fraction of frequency of segments in the subwindow tm, as determined by Formula (4).

$$w_t(s, m) = f_t(s, m) / f_{s,t} \quad (4)$$

Using the similarity measure provided in Formula (3), all bursty segments undergo clustering utilizing a modified version of the Jarvis-Patrick algorithm (Jarvis and Patrick, 1973). Every segment is first regarded as a node, and none of the nodes are initially connected. If s_b is one of s_a's k-nearest neighbors, then there is an edge between s_a and s_b, and vice versa.

Candidate event clusters are represented by the connected components of the graph after all possible edges have been added. Segments devoid of any edges are eliminated from further investigation.

However, post-clustering, it was observed that certain clusters were unrelated to any specific event. A cluster containing elements such as "[sunday dinner]," "[sunday night]," "[every sunday]," "[sunday funday]," and "[next sunday]" was discovered via Sunday, October 14, 2012, tweets. This cluster indicated recurrent events on particular days of the week.

To address this issue, filtering is necessary to eliminate such non-event clusters, prompting the utilization of an external knowledge base such as Wikipedia.Formula (5) defines the newsworthiness(s) of a segment s. This means that if a segment's sub-phrase is important, the segment will be considered newsworthy.This measure helps in distinguishing segments with substantive content from those with less relevance.

$$\mu(s) = \begin{cases} e^{Q(s)}, & \text{if } s \text{ is a word} \\ \max_{l \in s} e^{Q(l)} - 1, & \text{otherwise} \end{cases} \quad (5)$$

Formula (6) defines the event cluster's newsworthiness e by considering the weight of the event cluster's edges, represented as segment similarity, as well as the newsworthiness of its individual segments. This measure considers the probability of each sub-phrase aiding in the filtration non-event clusters measure takes into account the likelihood that each subphrase l inside a segment's' will show up as anchor text in l-containing Wikipedia pages. These elements are combined to determine an event cluster's newsworthiness by weighing.

$$\mu(e) = \left(\Sigma_{s \in e_s} \mu(s)\right) / |e_s| * \left(\Sigma_{g \in E_e} sim(g)\right) / |e_s| \quad (6)$$

E_s represent the set of segments related to event e, the set of edges connecting segments within event e by E_e, and the similarity between nodes of the edge g by sim(g), which is determined using Formula (3), the connections within the cluster, and the significance of its individual segments. It has been observed that candidate events lacking realistic potential tend to exhibit very low newsworthiness values compared to genuine events. Therefore, only events ‘e’ satisfying the condition e_s > T are retained as realistic events, while others are discarded. In this case, T stands for a threshold value, and \(max) indicates the highest newsworthiness among all potential event clusters.

This filtering mechanism ensures that only events with significant newsworthiness are considered realistic, thereby enhancing the accuracy of event detection.

D. Event Summarization

Recognizing that a mere list of segments associated with an event cluster may not encompass all pertinent information regarding an event, we applied the LexRank algorithm (Erkan and Radev, 2004) to generate summaries of the event clusters identified in the preceding step. The LexRank algorithm operates by synthesizing top-ranking sentences from multiple documents to create a concise summary. In our setup, an event is summarized using any tweets from the current time window (let "t") that contain the segments from the event cluster.

The segment index that was produced during the tweet segmentation stage has these sections. Using the LexRank algorithm, we are able to extract a detailed account of the event from this set of tweets, encompassing both its salient features and subtleties.

III. RESULTS

This section will provide the evaluation measures that were used, the dataset that was used, and some statistics related to tweet segmentation and our conclusions. Our model surpasses Twevent (Li et al., 2012a) in terms of precision, yielding a higher number of events while minimizing duplicate occurrences.

A. Experimental Setting and Dataset

Subsection 2.1's collection of Wikipedia page titles was compiled from a dump that was acquired in March 2018 and had 8,007,358 page titles in total. For calculating the Wikipedia key-phrases values Q(s), we utilized the dataset utilized by Li et al. (2012a), which was based on a dump released on January 30, 2010. 4,342,732 different things that appear in an anchor text were included in this dataset.

A Twitter corpus called Events2012 was constructed by McMinn et al. (2013) and includes tweets from October 10 to November 7, 2012.

They applied filtering criteria to remove tweets with excessive hashtags, name mentions, or URLs, as they may indicate spam (Benevenuto et al., 2010). Following this preprocessing, there were almost 120 million tweets in the corpus, along with a list of 506 events broken down into 8 categories. This corpus was used to assess the performance of our model and estimate probability of segments p_s, as discussed in part 2.2. PyTweetCleaner was used to preprocess the tweets from the corpus as well as the Wikipedia Page Titles dataset.The size of the time window and the number of sub-windows M were altered, hashtag weight H, threshold T, number of neighbors k during clustering, and many parameters to assess the model's effectiveness.

The time window is 24 hours long, divided into M = 12, 2-hour sub-windows. Furthermore, we set up T = 4, H = 3, and k = 3 neighbors.Precision, defined by Allan et al. (1998) as "the fraction of detected events related to realistic events," and Duplicate Event Rate (DERate), defined by Li et al. (2012a) as "the percentage of events detected more than once among all realistic events detected," were used as evaluation metrics. We refrained from using recall due to the absence of an exhaustive event list in the Events2012 dataset (McMinn et al., 2013). Our model, SEDTWik, identified 48 events within the period of October 11 to October 17, 2012, that were not reported by McMinn et al. (2013), as confirmed by their subsequent work (McMinn and Jose, 2015).

Table 1 illustrates some events detected by SEDTWik that were missed by McMinn et al. (2013). Please take note that the generated summaries are too long to fit inside the table, so the event information is manually supplied. To assess SEDTWik's performance, we use the number of events detected as a metric rather than recall.

Table 1. A few occurrences that SEDTWik identified between October 11 and October 17, 2012, but which McMinn et al. (2013) did not discover

Date	Event Info
11 Oct 2012	The mechanics of a plane accident will be studied in real time using a Boeing 727 passenger airplane.
12 Oct 2012	Tennessee Titans vs. Pittsburgh Steelers football game.
13 Oct 2012	Rylan Clark and James Arthur, X Factor UK finalists, live in London.
14 Oct 2012	Football game between the South Carolina Gamecocks and LSU for the National Championship.
15 Oct 2012	Ray Lewis may have missed a year or even his career due to a torn tricep muscle.
16 Oct 2012	President Barack Obama reminds Mitt Romney at the US presidential debate that he is the last to criticize China harshly.
17 Oct 2012	Tweets on planned parenting and birth control.

B. Tweet Segmentation Statistics

We removed all retweets from 11,705,978 tweets that were evaluated during the week of October 11–October 17, 2012. There were 3,653,039 distinct segments in this dataset. The distribution of segment lengths and their frequencies for this time period is shown in Figure 3. Many of the bigrams, as we saw, were coherent statements like "passed away" or proper nouns like [nicki minaj], [mitt romney]. Table 2 shows examples of tweet segmentations with a hashtag weight of 3. Pop singer Demi Lovato is linked to the account @ddlovato in the second row.

Table 2: Example Of Tweet Segment

Tweet	Segments
im pretty sure keyshawn johnson new favorite analyst #NFL ESPN	#NFL ESPN[new favorite], [Keyshawn Johnson], [NFL ESPN]

got tds season last game next week in varsity guy @ LaQuon Treadwell	[last game], [LaQuon Treadwell]
stasik agnes berlin nj needs wedding florist #Wedding #BrideToBe #Caterer #Videographer	[wedding florist], [Wedding], [BrideToBe], [Caterer], [Videographer]

Table 3: Twevent and our method, SEDTWik, are compared for events that occurred between October 11 and October 17, 2012.

Approach	Number of events	Accuracy
SEDTWik	79	88%
Twevent	42	80%

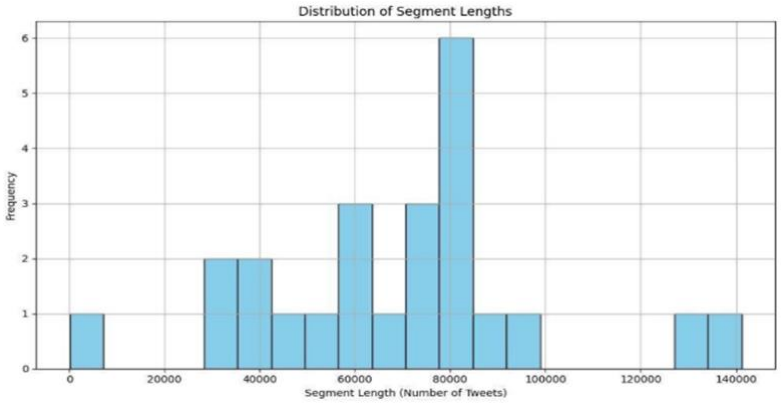

Fig. 2 Tweet count for each file

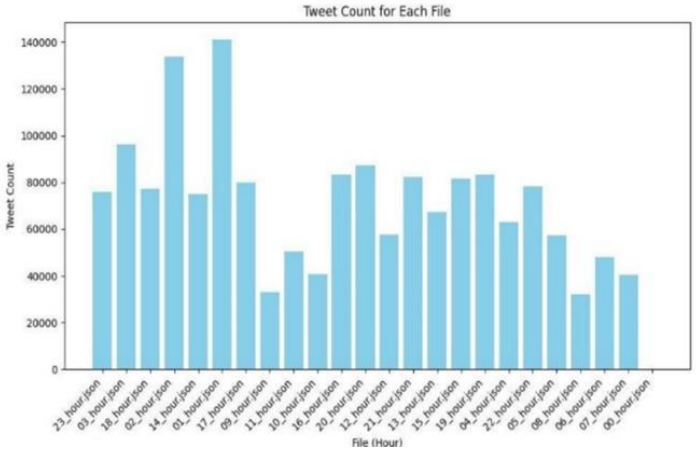

Fig.3 Segmentation distribution

C. Event Detection and Results

Twevent, a method proposed by Li et al. in 2012, is designed for detecting events in tweets using tweet segmentation and proved to be more effective than EDCoW (Weng and Lee, 2011), which was the top-performing method at the time. Precision, recall, quantity of events recognized, and reduced duplicate rates were all excellent. We compare Twevent's results in this section since SEDTWik, our model, is a Twevent extension. Table 3 contrasts SEDTWik and Twevent in terms of accuracy, duplication rate (DERate), and the number of events

discovered for tweets sent between October 11 and October 17, 2012. Note that we do not measure recall because there isn't a comprehensive list of events for that period We computed results for Twevent and our model using our own probability estimates, rather than the defunct Microsoft Web N-gram service. We manually marked the event clusters as real or not once they were created and summaries were generated, and we then computed precision.

SEDTWik achieved a precision of 88.12%, higher than Twevent's 80.32%, as shown in Table 3. Additionally, it detected a lot more events—79 as opposed to 42 from Twevent. SEDTWik likewise outperformed Twevent in terms of DERate (14.10% vs. 16.67%). This shows that in all three categories, SEDTWik performs better than Twevent. Other studies, like Edouard et al. (2017) and TwitterNews+ (Hasan et al., 2016), also used the Events2012 dataset (McMinn et al., 2013) to evaluate their models, though for different periods. Their precision values were 75.0% and 78.0%, respectively. We didn't re-assess our model with these tweets because of the manual annotation required, but we believe our precision would be higher than theirs The top segments from each event cluster are listed in Table 4, along with a summary of some of the events that SEDTWik observed on a daily basis between October 11 and October 17, 2012. . The event descriptions are handwritten to fit the table because the summaries include a large number of tweets. The data, the code, and the full SEDTWik project are available for replacing the username with the corresponding full name, the segment becomes easier to understand.

Table 4. The upper portions of the event cluster and a number of the occurrences that SEDTWik discovered for each day between October 11 and October 17, 2012.

Date	Event
Oct 11	[mo yan], [chinese writer], [nobel prize literature] -> The Nobel Prize in Literature is given to Chinese author Mo Yan. [national coming out day], [lgbt]→ On this day, people celebrate National Coming Out Day.
Oct 12	[nobel peace prize], [european union]-> The Nobel Peace Prize for 2012 goes to the European Union.
Oct 13	[xfactor], [x factor], [james arthur], [rylan clark] → Rylan Clark and James Arthur, finalists on X Factor UK, perform live in London.

D. Impact of H and T

The variable H was utilized in subsection 2.1 to ascertain the weight of hashtags in the tweet segmentation process. The value of H indicates the multiplier for the frequency of hashtags. Since users typically use hashtags to highlight important aspects of tweets, and these hashtags are often common across related tweets, it makes sense to assign them a higher weight. A low value for H, however, could cause noise from typical tweet text to overwhelm the segmentations and reduce the event detection model's accuracy.

Likewise, a high H value would suppress other frequently occurring segments, which would decrease accuracy.We tested with H values of 1, 2, 3, and 4, and found that H = 3 produced the greatest results.

The threshold T, as stated in subsection 2.3, was applied to determine if a candidate cluster qualified as an actual event.We found that increasing T led to more clusters being classified as events, thus increasing the overall count of detected events, but reducing the model's precision. By experimenting with different T values (2, 3, 4, and 5), we determined that T = 4 provided optimal results

E. Comparative Analysis with Elbow Method + NMF

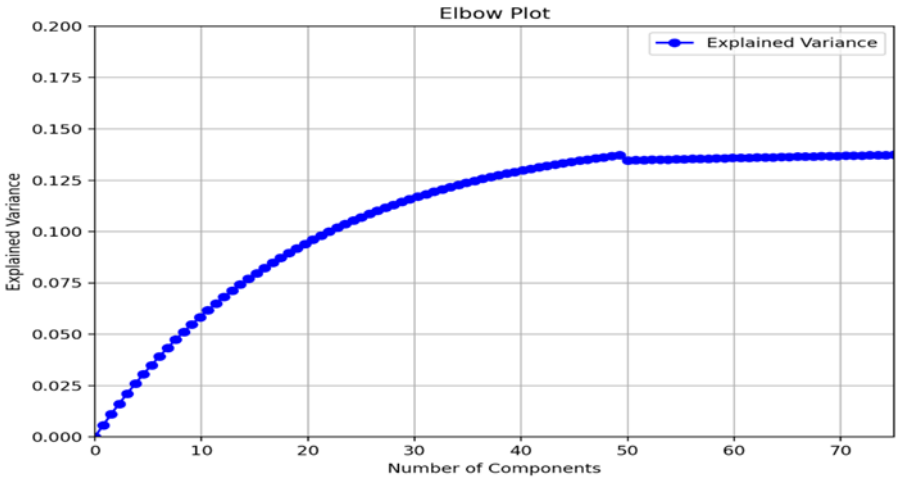

Fig. 4 Using the Elbow Method to Determine the Ideal Component Count

To assess the effectiveness of SEDTWik, we compared its performance with the Elbow Method combined with Non-Negative Matrix Factorization (NMF). After extracting tweets from JSON and figuring out the ideal number of clusters using the Elbow Method, I observed a gradual increase in explained variance without a clear "elbow" point, making the selection of the number of components subjective.

We proceeded with 50 components and applied NMF, which resulted in significant overlap and redundancy across topics. For instance, Topics 5, 19, and 43 were all centered around debates but failed to distinguish between specific debate events, while Topics 7, 8, and 9 contained generic terms such as "got," "wait," and "night," reflecting everyday conversations rather than distinct events. In contrast, SEDTWik outperformed NMF by detecting 79 meaningful events with higher granularity, reducing redundancy (lower DERate), and achieving a precision of 88.12%, far surpassing NMF's capabilities in capturing finer details. The temporal segmentation integrated within SEDTWik also allowed it to identify events occurring at similar times but in different contexts, something NMF struggled to capture. These advantages highlight SEDTWik's superior ability to detect specific sub-events in social media datasets, providing more accurate and contextually relevant event detection.

Table 5. Some Topics Extracted from Tweets using NMF

Topic number	Topic
1	Surveys and waiting for responses this week
5	Discussion on Biden, Ryan, and the debate
10	Waiting for Biden's response and opinions on the situation
16	Anticipation for Biden's actions tomorrow

IV. Related Work

Over the past 10 years, event detection in tweets has been the subject of much research; this section summarizes some of the seminal papers that have influenced our own work in this area. Panagiotou et al. (2016), Weiler et al. (2016), and Farzindar and Khreich (2015) conducted comprehensive surveys of event detection techniques used in Twitter-based research. They also outlined various challenges in this domain, which our work aims to address.

TwInsight (Valkanas and Gunopulos, 2013) tracked emotional spikes across six states—anger, fear, disgust, happiness, sadness, and surprise—to identify events, providing details like location, timestamp, and escriptions. EvenTweet (Abdelhaq et al., 2013) detected events by identifying words with a burstiness level of at least two standard deviations above the mean, and then clustering these words. This approach focused on local events. Similarly, EventRadar (Boettcher and Lee, 2012) used Twitter to identify events in specific areas, such as parties and art shows.

McMinn et al. (2013) clustered events using a combination of temporal, category-based, and content-based features using Locality locality-sensitive hashing (LSH), resulting in events categorized into eight groups. Science & Technology, sports, and other topics. However, as Table 1 shows, their method missed several events. Phuvipadawat and Murata (2010) employed features like hashtags, usernames, follower counts, retweet counts, and proper noun terms to cluster and rank breaking news events detected from Twitter.

Twevent (Li et al., 2012a) applied a segmentation-based approach to event detection, where segments were rated by "stickiness," and bursty segments were selected based on prior probability distribution and user diversity, then grouped into event clusters. We extend Twevent to our SEDTWik model; in part 3.3, we compare these approaches. s

More recently, ArmaTweet (Tonon et al., 2017) applied semantic event detection to tweets, allowing it to identify events like 'politician dying' and 'militia terror act.'

V. Conclusion And Future Work

Twitter possesses seen a rapid surge in user numbers and the volume of content, drawing significant attention from both industry and academia. However, the brevity of tweets and the presence of noisy data in high volume make event detection challenging. We present an event detection system in this work called SEDTWik based on tweet segmentation that leverages user popularity, follower counts, retweet counts, and hashtags. Enhancing the weight assigned to hashtags improved the system's efficiency considerably. Our model produced excellent results in detecting events from tweets, using Wikipedia as a reference.

In the future, we want to enhance the segmentation procedure and add URL linkages to event detection. We also aim to develop more accurate methods for estimating segment probabilities, taking into account the specific days and months when certain segments appear. Furthermore, instead of only extracting tweets for summary, we will focus on advanced event summarizing techniques that utilize segment and cluster data.

References

[1] D. Kothadiya, C. Bhatt, K. Sapariya, K. Patel, A.-B. Gil-González, and J. M. Corchado, “Deepsign: Sign language detection and recognition using Deep Learning,” *Electronics*, vol. 11, no. 11, p. 1780, Nov. 2022. doi: 10.3390/electronics11111780.

[2] B. D. Khuat, T. T. Nguyen, T. T. Nguyen, and Q. T. Nguyen, “Vietnamese sign language detection using Mediapipe,” in *Proc. 10th Int. Conf. on Software and Computer Applications*, 2021, pp. 3457784–3457810. doi: 10.1145/3457784.3457810.

[3] [3] V. Adithya, P. R. Vinod, and U. Gopalakrishnan, “Artificial neural network based method for Indian sign language recognition,” in *2013 IEEE Conf. on Information and Communication Technologies*, 2013, pp. 315–319. doi: 10.1109/cict.2013.6558259.

[4] P. S. Rajam and G. Balakrishnan, “Real-time Indian sign language recognition system to aid deaf-dumb people,” in *2011 IEEE 13th Int. Conf. on Communication Technology*, 2011, pp. 6157974. doi: 10.1109/icct.2011.6157974.

[5] V. Chunduru, A. R. Hande, G. M. Varghese, and M. M., “Hand tracking in 3D space using MediaPipe and PNP method for intuitive control of Virtual Globe,” in *2021 IEEE 9th Region 10 Humanitarian Technology Conf. (R10-HTC)*, 2021, pp. 9641587. doi: 10.1109/r10-htc53172.2021.9641587.

[6] I. Papastratis, G. Vouros, and K. Milios, “Artificial Intelligence technologies for sign language,” *Sensors*, vol. 21, no. 17, p. 5843, Aug. 2021. doi: 10.3390/s21175843.

[7] D. Parida, A. Panda, J. Rangani, and A. K. Parida, “Real-time environment monitoring system using ESP8266 and ThingSpeak on Internet of Things platform,” in *2019 Int. Conf. on Intelligent Computing and Control Systems (ICCS)*, 2019, pp. 9065451. doi: 10.1109/iccs45141.2019.9065451.

[8] M. J. Cheok, Z. Omar, and M. H. Jaward, “A review of hand gesture and sign language recognition techniques,” *Int. J. Machine Learning and Cybernetics*, vol. 10, no. 1, pp. 131–153, Jan. 2017. doi: 10.1007/s13042-017-0705-5.

[9] T. Starner and A. Pentland, “Real-time American sign language recognition from video using Hidden Markov models,” in *Proc. Int. Symp. on Computer Vision - ISCV*, 1995, pp. 477012.

[10] M. A. Razali, M. H. Mohd, and M. F. Mohamed, “A ThingSpeak IoT on Real-Time Room Condition Monitoring System,” in *2020 IEEE Int. Conf. on Automatic Control and Intelligent Systems (I2CACIS)*, 2020, pp. 9140127. doi: 10.1109/i2cacis49202.2020.9140127.

[11] C. Li, A. Sun, and A. Datta, “Twevent: Segment-based event detection from tweets,” in *Proc. 21st ACM Int. Conf. on Information and Knowledge Management (CIKM '12)*, 2012, pp. 155-164. doi: 10.1145/2396761.2396785.

[12] G. P. C. Fung, J. X. Yu, P. S. Yu, and H. Lu, “Parameter free bursty events detection in text streams,” in *Proc. 31st Int. Conf. on Very Large Data Bases (VLDB '05)*, 2005, pp. 181-192.

[13] E. Amosse, E. Cabrio, S. Tonelli, and N. Le-Thanh, “Graph-based Event Extraction from Twitter,” in *Proc. Int. Conf. on Recent Advances in Natural Language Processing (RANLP 2017)*, 2017, pp. 222-230. doi: 10.26615/978-954-452-049-6_031.

[14] C. G. Puri, S. M. Rokade, and K. N. Shedge, “Segmentation of tweets and real-time event detection using NER & HybridSeg,” in *Nat. Level Conf. on Advanced Computing and Data Processing (ACDP 2K19)*, 2019.

[15] R. Patankar and A. Pravin, “A novel optimization-assisted multi-scale and dilated adaptive hybrid deep learning network with feature fusion for event detection from social media,” *Network: Computation in Neural Systems*, vol. 35, no. 4, pp. 429–462, Apr. 2024.

[16] R. A. Patankar and A. Pravin, “Event detection from social media using machine learning,” in *Lecture Notes in Electrical Engineering (LNEE, Vol. 690)*, Int. Conf. on Emerging Trends and Advances in Electrical Engineering and Renewable Energy, 2021, pp. 539–548.

[17] J. Wang and X.-L. Zhang, “Deep NMF topic modeling,” *Neurocomputing*, vol. 515, pp. 157–173, Jan. 2023. doi: 10.1016/j.neucom.2022.10.002.

[18] E. Umargono, J. Suseno, and S. K. Gunawan, “K-means clustering optimization using the elbow method and early centroid determination based on mean and median formula,” *Advances in Social Science, Education, and Humanities Research*, vol. 10, p. 019, 2020. doi: 10.2991/assehr.k.201010.019.

[19] S.-W. Kim and J.-M. Gil, “Research paper classification systems based on TF-IDF and LDA schemes,” *Human-centric Computing and Information Sciences*, vol. 9, Article 30, Nov. 2019. doi: 10.1186/s13673-019-0192-7.