MetroScope: An Advanced System for Real-Time Detection and Analysis of Metro-Related Threats and Events via Twitter

Jianfeng He
jianfenghe@vt.edu
Virginia Tech
Falls Church, VA, USA

Syuan-Ying Wu
syuanyingwu@vt.edu
Virginia Tech
Falls Church, VA, USA

Abdulaziz Alhamadani
hamdani@vt.edu
Virginia Tech
Falls Church, VA, USA

Chih-Fang Chen
chenchihfang@vt.edu
Virginia Tech
Falls Church, VA, USA

Wen-Fang Lu
wenfang@vt.edu
Virginia Tech
Falls Church, VA, USA

Chang-Tien Lu
ctlu@vt.edu
Virginia Tech
Falls Church, VA, USA

David Solnick
dsolnick@wmata.com
Washington Metropolitan Area Transit Authority
Washington, D.C., USA

Yanlin Li
yli@wmata.com
Washington Metropolitan Area Transit Authority
Washington, D.C., USA

ABSTRACT

Metro systems are vital to our daily lives, but they face safety or reliability challenges, such as criminal activities or infrastructure disruptions, respectively. Real-time threat detection and analysis are crucial to ensure their safety and reliability. Although many existing systems use Twitter to detect metro-related threats or events in real-time, they have limitations in event analysis and system maintenance. Specifically, they cannot analyze event development, or prioritize events from numerous tweets. Besides, their users are required to continuously monitor system notifications, use inefficient content retrieval methods, and perform detailed system maintenance. We addressed those issues by developing the MetroScope system, a real-time threat/event detection system applied to Washington D.C. metro system. MetroScope can automatically analyze event development, prioritize events based on urgency, send emergency notifications via emails, provide efficient content retrieval, and self-maintain the system. Our MetroScope system is now available at http://orion.nvc.cs.vt.edu:5000/, with a video (https://www.youtube.com/watch?v=vKIK9M60-J8) introducing its features and instructions. MetroScope is a significant advancement in enhancing the safety and reliability of metro systems.

CCS CONCEPTS

• **Human-centered computing** → **Visualization systems and tools**; **Social network analysis**.

KEYWORDS

Twitter analysis, database, visualization

SIGIR '23, July 23–27, 2023, Taipei, Taiwan

ACM ISBN 978-1-4503-9408-6/23/07.
https://doi.org/10.1145/3539618.3591807

ACM Reference Format:
Jianfeng He, Syuan-Ying Wu, Abdulaziz Alhamadani, Chih-Fang Chen, Wen-Fang Lu, Chang-Tien Lu, David Solnick, and Yanlin Li. 2023. MetroScope: An Advanced System for Real-Time Detection and Analysis of Metro-Related Threats and Events via Twitter. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '23), July 23–27, 2023, Taipei, Taiwan.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3539618.3591807

1 INTRODUCTION

Metro systems are vital to any major city, serving countless passengers every day. In the Washington D.C. area alone, the average daily ridership in 2019 was about 630,000 [24]. However, events or threats that occur in metro stations not only jeopardize the safety and convenience of passengers but can also tarnish the city's image. These incidents range from minor delays to criminal or terrorist activities. A real-time system that can detect and collect such threats or events is critical for metro authorities to take timely action. Moreover, by analyzing the development of previous threats/events, authorities can proactively prevent recurring threats/events, leading to a safer and more dependable service for passengers. Regrettably, most current metro-related systems, can only provide information on pre-set schedules or maintenance-related disruptions set by administrators. For example, Google Maps only provides the expected arrival time of the next metro without analyzing metro-related events. Hence, there is a pressing need for a system that can detect, analyze, and summarize metro-related threats/events in real-time, without depending on pre-set information.

Ensuring passengers' safety is crucial, but always having police in every metro is impossible due to limited police resources. To simultaneously detect threats or events in many metros, one solution is to leverage passengers as the eyes of public safety [7]. Fortunately, social media platforms enable people to become "human sensors" for almost every metro. Among the social media platforms, Twitter is one of the most influential platforms, with 199 million daily active users and 500 million daily tweets in 2022 [3]. Although many

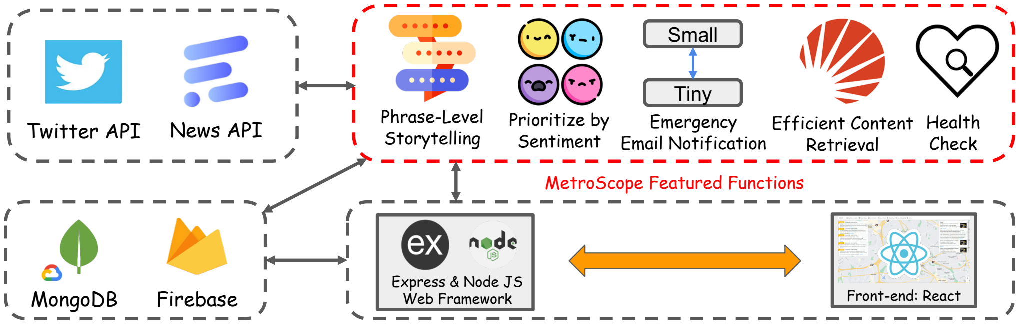

Figure 1: System overview of our MetroScope, where the red rectangle shows its featured functions.

works analyzed the traffic-related tweets [1, 16, 17, 20, 22], only a few implemented systems to detect and visualize threats/events from tweets in real-time. However, these implemented systems [8, 12, 18, 26] have limitations in event analysis and system maintenance. MetroScope is developed to address the limitations. It is a real-time threat/event detection system applied on the Washington D.C. metro system (WMATA). The limitations in current works and our contributions in MetroScope are below.

Analyzing event development by storytelling. To prevent recurrent incidents, authorities need to understand how events progress over time. However, current Twitter-based monitoring systems for metro-related events [8, 12, 18, 26] lack this function. Hence, we propose a phrase-level storytelling algorithm that groups events with similar key phrases into a story. Each story provides a clear and intuitive visualization of an event's progression, allowing authorities to better understand its development.

Prioritizing events using sentiment analysis. Although current systems can detect events and threats [8, 12, 18, 26], they lack the ability to rank them by priority. This means that authorities cannot address the most urgent events quickly. To prioritize the events by urgency, we implement a sentiment analysis module to calculate the user emotion of each event. This enables authorities to handle events with a greater public impact at first.

Real-time emergency notification using synonyms via email. Immediate notification of emergency events minimizes authorities' response time. However, current systems lack emergency notification capabilities [8, 12, 18] or require constant monitoring [26], which can be inconvenient for authorities. To address this, we implemented an email-based emergency notification function that emails notifications directly to authorities. We also added a synonym module to identify emergency-related keyword synonyms, reducing the likelihood of missing critical notifications. This new functionality provides authorities with a more efficient and convenient way to stay informed about emergency events in real-time.

Improving Content Retrieval Efficiency. Current systems like [26] rely on string matching for keyword-based retrieval, which is time-consuming. To improve efficiency, we have implemented an index-based keyword retrieval method using Solr [13]. This enables faster retrieval of content related to specific keywords.

Implementing a system health-check module. Maintaining multi-functional systems can be challenging due to difficulty in identifying failed modules. Current systems lack this function [8, 12, 18, 26]. To solve this, we created a system health-check module that detects anomalies in the system's functions. It automatically restarts any failed functions, saving time and money on maintenance while improving the system's reliability.

2 SYSTEM DESCRIPTION

In this section, we present the architecture of MetroScope. We will first provide an overview of the system, followed by a detailed explanation of each new function.

2.1 System Overview

As shown in Fig. 1, MetroScope collects tweets and news by Twitter API (Tweepy API [23]) and news API (Rapid [19] And News [15] APIs), respectively. We only collect tweets and news that are related to metro-related keywords, such as names of metro stations. The collected tweets and news are saved into either a MongoDB [4] or a Firebase [14] database. Then, the collected twitters and news are initially displayed in our frontend. For example, MetroScope updates its real-time panel once a new tweet or recent news is detected. Once a user makes a request from the frontend to the backend, the backend then finishes the request with the help of our databases and our implemented functions. The finished results are then returned to the frontend for a visualization update. Compared to current twitter-based metro-related systems [8, 12, 18, 26], MetroScope has five new functions: "storytelling" for summarizing an event development, "sentimental analysis" for prioritizing the events, "emergency email notification" (EEN) for convenient notifications, "content retrieval" for efficient keyword retrieval, and "system health-check" for easy maintenance.

2.2 Function 1: Storytelling

To prevent recurrent threats and events, it is important for authorities to understand the development of an event. With this in mind, MetroScope uses our proposed storytelling algorithm to collect events with similar key phrases as a story, shown in the blue dashed rectangle in Fig. 3. To quickly understand a story, we extract keywords from each story. Because there could be several stories in the same period, we list their respective keywords in the red dashed rectangle in Fig.3 to provide a global view of the stories. Plus, a time range selection panel in the orange rectangle in Fig. 3 allows a user to obtain stories from any day to its past 24 hours.

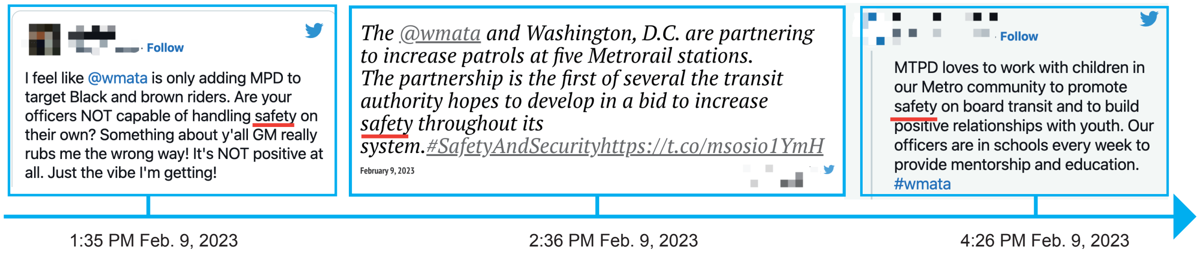

Figure 2: Case study of storytelling, where the example shows three safety-related events that happened in one day.

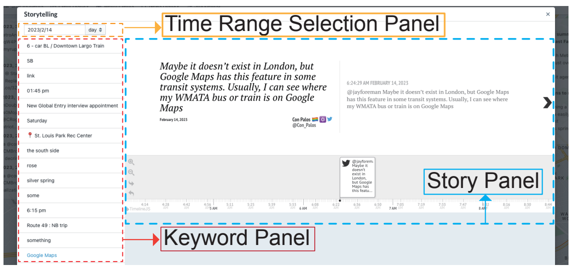

Figure 3: A screenshot of the storytelling function.

For our storytelling algorithm, we first gather n tweets that are posted within a selected time range. Then, for i-th tweet, we extract l_i key phrases $P_i = [p_i^1, p_i^2, ..., p_i^{l_i}]$ via a pretrained event extraction model from Spacy [21]. For all key phrases $\mathbf{P} = [P_1, P_2, ..., P_n]$ from n tweets, we apply DBSCAN clustering [9] to group them into k clusters. Then, because the phrases in each cluster are similar, their corresponding tweets are related. However, some phrases in each cluster are commonly used, such as "we" and "metro", and lead to trivial stories. To address this, we calculate importance scores [2, 5] for phrases $\hat{\mathbf{P}}$ that are extracted from one-year tweets, as below.

$$s(p_j) = \frac{\epsilon}{\epsilon + U(p_j)} \tag{1}$$

where $\epsilon = 0.001$, p_j is j-th phrase among $\hat{\mathbf{P}}$, and $U(p_j) \in [0, 1]$ is the unigram likelihood of p_j over the $\hat{\mathbf{P}}$. We then filter out key phrases among the k clusters that have importance scores lower than a pre-set threshold of 0.51. Finally, we select the remaining key phrases in each cluster and use their corresponding tweets to create a story (blue rectangle in Fig.3), while the set of remaining key phrases serves as the story keywords (red rectangle in Fig.3).

2.3 Function 2: Prioritize Tweets by Sentiment

When many events occur concurrently, it is essential for authorities to prioritize urgent events. To assist with this, we have added a sentiment analysis function to each tweet, enabling authorities to process the most urgent events with higher priority. For sentiment analysis, we utilize VADER [10], a dictionary containing human-annotated phrase-emotion pairs. VADER returns a compound emotion score for a given text input, ranging from -1 to 1. Using the compound emotion score e of a tweet, we classify it into one of seven different emotion categories, Extremely Negative $(-1.0 \le e < -0.7)$, Very Negative$(-0.7 \le e < -0.4)$, Fairly Negative $(-0.4 \le e < -0.1)$, Neutral $(-0.1 \le e \le 0.1)$, Fairly Positive $(0.1 < e \le 0.4)$, Very Positive $(0.4 < e \le 0.7)$ and Extremely Positive $(0.7 < e \le 1.0)$. With the sentiment analysis, every crawled tweet is assigned an emotion category before being saved to the database. Furthermore, we have implemented a sentiment filter to choose tweets with a particular emotion category. For instance, as shown in Figure 4, MetroScope can display only tweets with "extremely negative" emotions by our sentiment filter as an example.

2.4 Function 3: Emergency Email Notification

Some metro systems [26] have frontend notifications for emergency events, but authorities must always monitor the screen to respond immediately. This is inconvenient. To address this issue, we propose implementing an Emergency Email Notification (EEN) function that automatically emails emergency events to authorities, enabling them to receive notifications without the need of constant monitoring. To implement the EEN function, administrators first set emergency event keywords in their accounts. Then, whenever MetroScope saves a newly crawled tweet to the database, it uses a string matching method to check whether any emergency event keywords appear in the tweet. If a match is found, the backend emails the tweet to the authorities, who can receive notifications via their mobile email app from anywhere.

Synonyms Module. Manual-set emergency event keywords in the EEN system can result in missed events since variations or synonyms of keywords cannot be matched. For instance, a tweet that mentions "firearm" cannot be matched if the keyword set only includes "gun." To address this issue, we have incorporated a synonyms module into the EEN system (as shown in Figure 5). This module leverages WordNet [6], which provides synonyms for a given keyword. By using this module, we can expand the set of emergency event keywords, reducing the likelihood of missed events. As a result, MetroScope can email more emergency notifications.

2.5 Function 4: Efficient Content Retrieval

Conventional content retrieval methods use string matching for keyword queries, which can be time-consuming and inefficient. To address this issue, we have implemented an efficient keyword retrieval system that uses index matching via Solr [13]. Solr creates an inverted index that maps words/characters to document IDs, where the documents contain the corresponding word/character. This enables faster retrieval compared to conventional string matching for

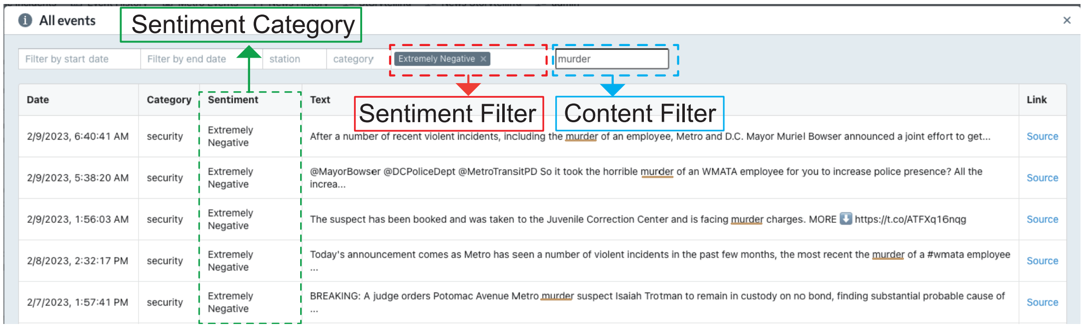

Figure 4: A screenshot for prioritizing tweets by sentiment "extremely negative" and content retrieval of "murder".

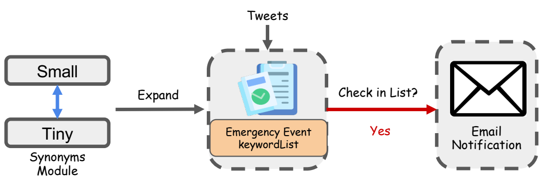

Figure 5: The workflow of emergency email notification.

keyword retrieval. Without Solr, a full scan of the MongoDB database is required to find matching documents. With Solr, MetroScope maps each tweet to its corresponding document IDs at the word and character levels. The document IDs for each tweet are saved in a Solr server. When querying by a keyword (as shown in Figure 6), Solr returns its document IDs to the backend, which can then retrieve the relevant tweets with their corresponding document IDs from the tweet database (such as MongoDB).

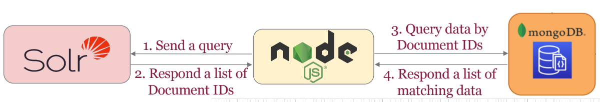

Figure 6: The workflow of efficient content retrieval.

2.6 Function 5: Health Check

When a system has multiple functions, it is difficult to identify the root cause of an issue without individually checking each function. To address this, MetroScope has implemented a health check module to automatically detect and restart any malfunctioning functions, saving time and maintenance costs. Specifically, MetroScope's storytelling, sentiment analysis, and EEN functions are all managed by Jenkins [11], a script manager. If Jenkins fails, these three functions will not run normally. To monitor Jenkins, we have set up Crontab [25], a job scheduler to check Jenkins' status every 5 minutes. If a Jenkins failure occurs, our health check module will automatically restart Jenkins and its associated functions.

3 CASE STUDY

Storytelling. In Fig. 2, we show a safety-related story consisting of three event screenshots. In the first tweet, a concern is raised about the ability of WMATA officers to handle safety. The second tweet introduces the measures taken by WMATA to increase safety, while the last tweet praises WMATA's safety measures. Our storytelling function enables authorities to easily summarize the progression of each event, aiding in the planning to prevent recurrent incidents.
Sentiment Analysis & Content Retrieval. The two functions are designed to be used separately, but Fig. 4 demonstrates a case study using them in conjunction. Specifically, we select tweets with sentiment categories that are all "extremely negative" and contain the keyword "murder." By combining these two functions, authorities can quickly identify and prioritize critical events, enabling them to respond more efficiently and retrieve relevant content faster.

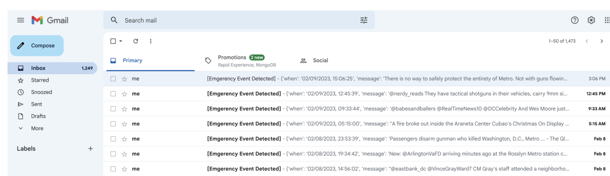

Figure 7: A screenshot of received emergency emails.

Emergency Email Notification. Fig. 7 shows our EEN function detecting tweets with the keyword "gun" in a screenshot of an authority's Gmail. This confirms that authorities can receive notifications on their mobile phones via email apps from anywhere, providing a fast and accessible means of monitoring emergencies.
Health Check. We tested the "health check" function by manually stopping any of the three functions, and found that it was automatically restarted, demonstrating the effectiveness of the function.

4 CONCLUSION

Detecting, collecting, and analyzing metro-related events are critical to passengers' safety and experience. While Twitter serves as a useful tool to gather information from "human sensors" in metro stations, currently implemented metro systems using tweets have issues with event analysis and maintenance. To overcome these issues, we developed MetroScope, featuring five new functions: "storytelling" for event development outlines, "prioritize tweets by sentiment" for events prioritization, "emergency email notification" (EEN) for convenient notifications, "content retrieval" for efficient keyword retrieval, and "health check" for easy maintenance. Our case studies demonstrated the effectiveness of these functions.

REFERENCES

[1] Apoorv Agrawal and Paulose N Kuriakose. 2022. Implications of a Twitter data-centred methodology for assessing commuters' perceptions of the Delhi metro in India. *Computational Urban Science* 2, 1 (2022), 38.

[2] Sanjeev Arora, Yingyu Liang, and Tengyu Ma. 2017. A simple but tough-to-beat baseline for sentence embeddings. In *International conference on learning representations.* 1–16.

[3] Salman Aslam. 2022. Twitter by the numbers (2022): Stats, demographics & fun facts. *Omnicoreagency. com* (2022).

[4] Kyle Banker, Douglas Garrett, Peter Bakkum, and Shaun Verch. 2016. *MongoDB in action: covers MongoDB version 3.0.* Simon and Schuster.

[5] Yujia Bao, Menghua Wu, Shiyu Chang, and Regina Barzilay. 2020. Few-shot Text Classification with Distributional Signatures. In *International Conference on Learning Representations.* 1–24.

[6] Keith Brown. 2005. *Encyclopedia of language and linguistics.* Vol. 1. Elsevier.

[7] Bertrand De Longueville, Robin S Smith, and Gianluca Luraschi. 2009. " OMG, from here, I can see the flames!" a use case of mining location based social networks to acquire spatio-temporal data on forest fires. In *Proceedings of the 2009 international workshop on location based social networks.* 73–80.

[8] Sri Krisna Endarnoto, Sonny Pradipta, Anto Satriyo Nugroho, and James Purnama. 2011. Traffic condition information extraction & visualization from social media twitter for android mobile application. In *Proceedings of the 2011 International Conference on Electrical Engineering and Informatics.* IEEE, 1–4.

[9] Martin Ester, Hans-Peter Kriegel, Jörg Sander, Xiaowei Xu, et al. 1996. A density-based algorithm for discovering clusters in large spatial databases with noise.. In *KDD*, Vol. 96. 226–231.

[10] Clayton Hutto and Eric Gilbert. 2014. Vader: A parsimonious rule-based model for sentiment analysis of social media text. In *Proceedings of the international AAAI conference on web and social media*, Vol. 8. 216–225.

[11] Jenkins. 2023. *Jenkins User Documentation.* https://www.jenkins.io/doc/

[12] Meiling Liu, Kaiqun Fu, Chang-Tien Lu, Guangsheng Chen, and Huiqiang Wang. 2014. A search and summary application for traffic events detection based on twitter data. In *Proceedings of the 22nd ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems.* 549–552.

[13] Nikola Luburić and Dragan Ivanović. 2016. Comparing apache solr and elasticsearch search servers. In *Proc. of the 6th International Conference on Information Society and Technology (ICIST 2016).* 287–291.

[14] Laurence Moroney and Laurence Moroney. 2017. The firebase realtime database. *The Definitive Guide to Firebase: Build Android Apps on Google's Mobile Platform* (2017), 51–71.

[15] NewsAPI. 2023. *Search worldwide news with code.* https://newsapi.org/

[16] Mohammed Ahsan Raza Noori and Ritika Mehra. 2021. Traffic congestion detection from twitter using word2vec. In *ICT Analysis and Applications: Proceedings of ICT4SD 2020, Volume 2.* Springer, 527–534.

[17] Joaquín Osorio-Arjona, Jiri Horak, Radek Svoboda, and Yolanda García-Ruíz. 2021. Social media semantic perceptions on Madrid Metro system: Using Twitter data to link complaints to space. *Sustainable Cities and Society* 64 (2021), 102530.

[18] Prabu Kresna Putra, Rahmad Mahendra, and Indra Budi. 2022. Traffic and road conditions monitoring system using extracted information from Twitter. *Journal of Big Data* 9, 1 (2022), 65.

[19] Rapid. 2023. *Millions of developers use the RapidAPI platform to build, use, and share thousands of APIs.* https://rapidapi.com/

[20] Kim Venitta Scott. 2019. *The Impact of Negative Twitter Feeds on Washington Metropolitan Area Transit Authority Ridership.* Hampton University.

[21] Bhargav Srinivasa-Desikan. 2018. *Natural Language Processing and Computational Linguistics: A practical guide to text analysis with Python, Gensim, spaCy, and Keras.* Packt Publishing Ltd.

[22] Mohhammad Sujon and Fei Dai. 2021. Social media mining for understanding traffic safety culture in washington state using twitter data. *Journal of Computing in Civil Engineering* 35, 1 (2021), 04020059.

[23] Twitter. 2023. *Twitter API v2.* https://docs.tweepy.org/en/stable/api.html

[24] WMATA. 2020. Metrorail ridership grew by 20,000 trips per weekday in 2019. https://www.wmata.com/about/news/2019-Metrorail-ridership.cfm.

[25] Lei Xu, San Xing Cao, and Li Ming Ma. 2014. A Strategy of Database Project Task Based on the Crontab. In *Applied Mechanics and Materials*, Vol. 610. Trans Tech Publ, 611–614.

[26] Omer Zulfiqar, Yi-Chun Chang, Po-Han Chen, Kaiqun Fu, Chang-Tien Lu, David Solnick, and Yanlin Li. 2020. RISECURE: metro incidents and threat detection using social media. In *2020 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM).* IEEE, 531–535.