Using Graph Evolutionary to Retrieve More Related Tweets

Patta Yovithaya* and Sukree Sinthupinyo

Abstract—**Due to its popularity and daily active users, social media has become powerful and influential in the last decade. With the nature of a micro-blogging platform, instant messages and the latest short posts are sent throughout the network on Twitter. Therefore, most users utilize Twitter to update breaking news or the latest events. Since a huge volume of tweet messages have been published on Twitter, event evolution has also rapidly developed into related events within similar topics. In this study, we present a novel method to retrieve tweets that relate to a given query term. Not only perfectly matched tweets, but more related tweets will be retrieved. The collected tweet data are processed and constructed as an original network. With the benefits of social network analysis, a simplification-based summarization approach is applied to ignore information that has less importance while preserving significant information in the network based on centrality measurement and clustering coefficient. Using the evolutionary of graph-based representation extends the relationship diffusion to assist related information retrieval. Experiments were performed using Thai news datasets and the framework performance was evaluated by precision, recall, and f-score. The experimental results show that our framework outperformed the baseline methods which derived a similarity score based on the word embedded vector to find relevant documents.**

Index Terms—**Graph evolutionary, graph summarization, information retrieval**

I. Introduction

With many active users of over 320 million and 500 million messages within a day [1], Twitter becomes one of the most popular social media platforms in the world. Twitter is a microblogging platform in which users are allowed to share either short blogs or instant messages by posting recent statuses or the latest news. With the nature of Twitter, the users could easily explore hot trending topics and instantly receive quick updates that happened around the world. A tweet does not only contain a creator message, but also includes thoughts and sentiments from different users related to its tweet. Wang *et al.* [2] studied Twitter users during the pandemic of COVID-19. They summed up the findings that comments & opinions, and news are the top reasons people use. Furthermore, a combination of news content and comments took the highest segment of all tweet diffusion. Since a huge number of users are actively generating several tweets every day, tweets are reproduced on many different topics. Therefore, there is a mix of vocabulary across Twitter's network every day. A typical representation that best fits both words and relations is a network graph which commonly consists of vertices and edges.

A graph or network is a representation composed of a collection of both vertices and edges. An edge is a link between two vertices. A graph typically visualizes the connection that interacts among vertices. With the time involved, vertices/edges can be added and deleted which is called a dynamic graph [3]. Social Network Analysis (SNA) is a field of graph analytical theory that aims to study a core network structure and interactions in the network. SNA has emerged in the past recent years and is widely studied in the diverse area of interest. To systematically perform analysis on a social network, the social network analysis metrics assist in many different aspects: discovering the pattern of connection, producing an underlying to differentiate networks, monitoring changes in a network over time, and analyzing locations in a network [4].

In this study, we utilize a simplification-based summarization technique that helps filter important nodes or edges from an original network to better understand the large social network. Furthermore, we exploit centrality measures which are one of the social network analysis theories to gain a deeper understanding of the network. To visualize the evolution of events over time, we introduce the evolution properties that provide added and overlapped vertices/edges in a consecutive day. The performance is evaluated by retrieving related tweets with a query term. Our proposed framework outperformed compared to document similarity-based approaches.

The details of the study are covered in the following sections. Section II includes related works. The process of our framework is described in Section III. In Section IV, we explain the experimental design used in this study. Section V provides the evaluation and experimental results. Lastly, Section VI concludes the paper.

II. Related Works

This section reviews works related to our study. Text summarization is a very popular technique for compacting the full-length original text into a concise summary while maintaining key important information. There are commonly two approaches in text summarization: extractive-based summarization and abstraction-based summarization. In the extractive-based summarization approach, the significant subsets of the original text will be pulled out and integrated to make a summary. On the other hand, an abstraction-based summarization is a technique of reproducing key crucial information using advanced techniques in a new shorter way. A variety of approaches are widely studied in graph-based text summarization. Mihalcea and Tarau authored a text summarization approach called TextRank [5] which is one of the most prominent algorithms where sentences represent the vertices of a graph and edges determine a similarity between sentences. Yongkiatpanich and Wichadakul [6] studied an

Manuscript received August 16, 2022; revised September 10, 2022; accepted November 2, 2022.

Patta Yovithaya and Sukree Sinthupinyo are with Department of Computer Engineering, Faculty of Engineering, Chulalongkorn University, Thailand.

*Correspondence: 6370220721@student.chula.ac.th (P.Y.)

DOI: 10.7763/IJCTE.2023.V15.1332

automatic extractive text summarization using PageRank and a combination of ontology and word embedding to improve the model that could determine a connection between sentences. Joshi *et al.* [7] authored a semantic graph-based text summarization approach for documents. They also used graph theory measurement to choose the summary sentences based on the top of their semantic scores. Natesh *et al.* [8] proposed a text summarization in a graph-based approach where the graph was built on a co-occurrence base. That method showed a good performance on news articles, Wikipedia searches, and technical documents. Recently, several studies have been proposed in abstraction-based summarization [9–11] and extractive-based summarization approaches [12–15].

We reviewed a survey paper to determine the summarization approaches in graph-based representation. Liu *et al.* [16] summarized the methods of graph summarization into four basis techniques: aggregation-based, bit compression-based, simplification-based, and influence-based. Our study focuses on a simplification-based approach that aims to pull out crucial vertices and edges from an original network. In other words, the core idea is to disconnect less important vertices and edges while preserving important information in a simplified network. Zeqian *et al.* [17] introduced a technique using the information of ontology to semantically prune the large networks called "OntoVis". The model supports both structural and semantic abstraction. In structural abstraction, the essential structure of the entire network is required to be preserved while pruning the network. Semantic abstraction allows users to construct a derived graph from the original graph by including only nodes whose types are selected in the ontology graph. Li and Lin [18] proposed an unsupervised algorithm for egocentric information abstraction in heterogeneous social networks that obtains similarly a resulting graph as OntoVis. Various recent studies [19–21] introduced how to simplify a network that includes discovering important vertices/edges with different metrics and pruning non-essential components of the network.

Several studies that researched the event evolution of social media platform. With a rapid change of events and opinions exchange, event evolution approaches emerged for detecting occurring events and discovering the evolution of events. Panagiotou *et al.* [22] summarized definitions, challenges, and trends in event detection frameworks that have been researched in the last few years. Cordeiro and Gama introduced a research survey of event detection techniques [23] that can generally be classified into two major methods. The first method is the online New Event Detection (NED) which points out to discovering events in real-time documents. The second technique, Retrospective Event Detection (RED), is a retrieval document process of event-relevant documents from the collection of historical documents. A discovery of event evolution from news stories was studied by Dou *et al.* [24]. The graph-based event evolution visualizes the core structure of events that can help the information extraction task and information retrieval.

Centrality measurement is one of the crucial SNA metrics that can determine how significant a vertex locates within the network. Vertex-specific network metrics provide several quantitative measures. Firstly, degree centrality measures the number of edges connected to a node [25]. The degree centrality of the vertex i is given in Eq. (1) where A_{ij} is the adjacency matrix of the graph and n is the number of vertices.

$$D(i) = \frac{1}{n-1}\sum_{j=1}^{n} A_{ij} \tag{1}$$

Betweenness centrality [26] is a measure that the extent of a node lies on the shortest path to other nodes. An equation is given in Eq. (2) where V is a set of nodes and $\sigma(s,t)$ is the number of shortest (s,t)-paths. Also, $\sigma(s,t|v)$ is the number of shortest $(s,\ t)$-paths passing through some node v other than s,t. If $s = t$, let $\sigma(s,t) = 1$, and if $v \in \{s,t\}$, let $\sigma(s,t|v) = 0$.

$$c_B(v) = \sum_{s,t\,\in\,V} \frac{\sigma(s,t|v)}{\sigma(s,t)} \tag{2}$$

Closeness centrality is a fraction of the sum of shortest path lengths to all other nodes given in Eq. (3). The notion of closeness centrality points up a node that can reach any other nodes in the graph either very far off or nearby with small hops [27].

$$C(u) = \frac{1}{\sum_y d(u,y)} \tag{3}$$

Clustering coefficient measures the density of connections in the network. The equation is given in Eq. (4) where $T(u)$ is the number of triangles through node u and deg (u) is the degree of u [28].

$$C_u = \frac{2T(u)}{\deg\,(u)(\deg(u)-1)} \tag{4}$$

III. Proposed Framework

Our proposed framework is comprised of five important procedures. We then start building an original network from the whole daily tweets where a particular word represents a node, and an edge displays a connection on adjacent words in a similar tweet message. A network is then constructed including words and relations, the next procedure is to measure three major centrality metrics and a clustering coefficient in the network. Only selected top N nodes of all indicators will be used to simplify the original network. Next, we extend further knowledge of how tweets evolve from one another by creating graph evolutionary properties. The steps to simplify a large network and create graph evolutionary properties are described in Algorithm 1.

```
Algorithm 1: Proposed framework
1. Collect tweets data using Twitter API
2. Preprocess collected data (remove special characters,
emoji, hashtags)
3. Tokenize each tweet into words
4. Construct a graph $G_i = (V, E)$ where a set of nodes $(V)$
represents the tokenized words and a set of edges
```

```
($E$) represents adjacent words in the same tweet
    4.1 Measure a degree centrality, betweenness centrality, closeness centrality, and clustering coefficient
    4.2 Sort all measurements in step 4.1 in descending order
5. Select unique nodes based on conditions:
    5.1 Clustering coefficient top N nodes and > 0
    5.2 Degree centrality top N nodes
    5.3 Betweenness centrality top N nodes
    5.4 Closeness centrality top N nodes
6. Simplify the graph based on selected nodes in step 5
     foreach $d_{i=1}$ in $D = \{d_{1,2,..,n}\}$:
         foreach $v$ in $V_d$:
             if $v$ not in selected nodes:
                 src ← neighbors of $v$
                 dst ← neighbors of $v$
                 new_edges ← cartesian_product(src, dst)
                 foreach $s, t$ in new_edges:
                   if $s \neq t$ :
                     $G_d$ ← add_edges_from($s, t$)
                     $G_d$ ← remove_node($v$ )
7. Create graph evolutionary properties
    foreach $d_{i=2}$ in $D = \{d_{1,2,..,n}\}$:
         $G_d$ ← Simplified graph of $d_i$
         $G_{d_{i-1}}$ ← Simplified graph of $d_{i-1}$
         added_nodes ← ($G_d$ nodes) – ($G_{d_{i-1}}$ nodes)
         overlapped_nodes ← ($G_d$ nodes) ∩ ($G_{d_{i-1}}$ nodes)
         added_edges ← ($G_d$ edges) – ($G_{d_{i-1}}$ edges)
         overlapped_edges ← ($G_d$ edges) ∩ ($G_{d_{i-1}}$ edges)
8. End of Algorithm
```

The overall system architecture, as illustrated in Fig. 1, shows how our framework processes a given query to find associated terms using graph evolutionary and retrieve related tweets. The detail of discovering expanded terms and retrieving related tweets will be described in the following section.

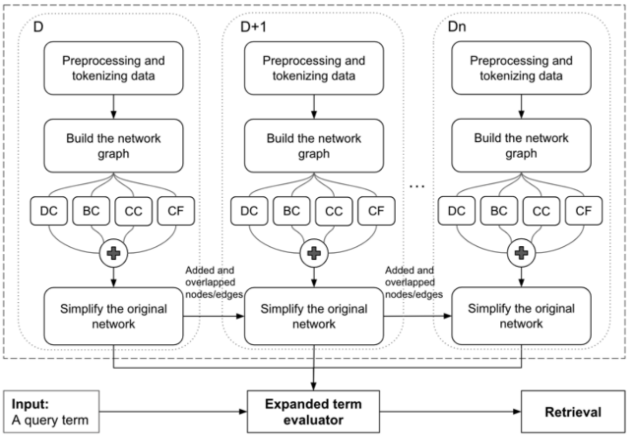
D
D+1
Dn
Preprocessing and tokenizing data
Build the network graph
DC
BC
CC
CF
...
Added and overlapped nodes/edges
Simplify the original network
Input: A query term
Expanded term evaluator
Retrieval

Fig. 1. The system architecture.

A. Building a Network Graph

We initially collect the tweets data using Twitter's Application Process Interface (API) [29]. Since users are allowed to post free-form texts, videos, and photos on Twitter, the data preprocessing process is mandatory. As this study focuses on text only, the collected data are cleansed to remove irrelevant information. Next, the step of tokenization is applied for splitting a tweet into words that represent vertices. The connections and relations become very crucial to capture the information association between words in the tweet, thus a pair of adjacent words in the same tweet represent the edges. A network will be constructed based on the collected daily tweet messages as shown in Fig. 2

B. Measuring Centrality and Clustering Metrics

Centrality is a significant measurement that shows the popularity, influence, and propagation in the network. Three centrality metrics measure how vertices play an importance in the network with different aspects. Firstly, Degree Centrality (DC) is a degree-based representative where edges are connected to a vertex. A high degree of centrality represents the vertex is hugely connected to any other vertices. A second measurement is Closeness Centrality (CC) which presents how an influential vertex can rapidly contact the others in small hops. In the other words, the vertices have the shortest path to the others and could be able to significantly propagate the information throughout the network. Third, a centrality that controls the information flow is Betweenness Centrality (BC). It aims to observe how important the vertex is when connected to the neighbors or acts as a center of communication among vertices or communities. Lastly, a measurement that reveals transitivity between neighboring vertices is introduced as a Clustering Coefficient (CF). A group of vertices close to a vertex is determined by computing the number of triangles around the vertex i normalized by the possible number of triangles [28].

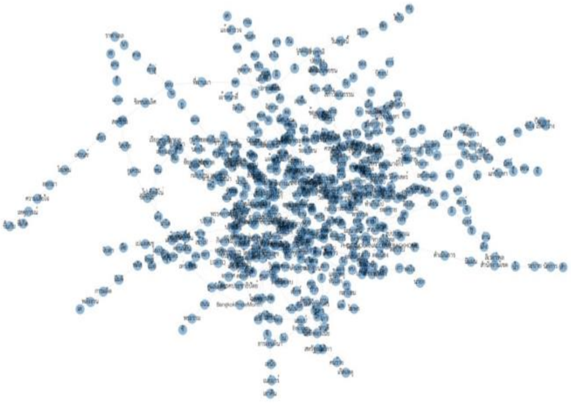

Fig. 2. The original network.

C. Simplifying a Network

Since the original network is measured using centrality and clustering metrics, high values give different benefits relying on a particular indicator. We merge top N vertices from DC, BC, CC, and CF into a set of selected vertices. The next process is to simplify the original network based on the top N vertices. A simplification-based graph is one of the graph summarization approaches that assist in filtering less important vertices and edges out of the network while keeping only important components. In exploring vertices that do not belong to a set of selected vertices, then simplification process is applied. First, finding neighbors of the vertex in the network. Then, we find the maximum possible edges among neighbors of the vertex that will be removed using the Cartesian product. After a set of new possible edges is created, each of them will be added to the network. Also, the vertex is removed from the network as shown in Fig. 3

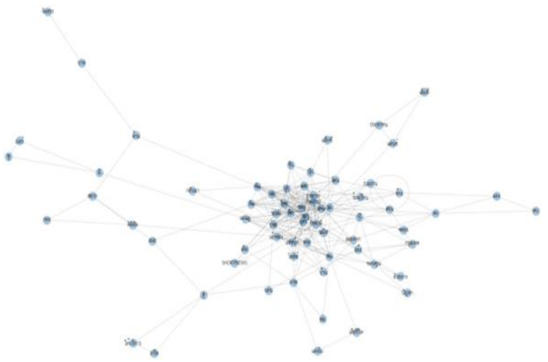

Fig. 3. The simplified network.

TABLE I: AN EXAMPLE OF GRAPH EVOLUTIONARY PROPERTIES

Added Nodes	territory, fill, warn, age, clue, register, welfare, advise, cafe, information, observe, islander, politician, genius, month, giant, beg, party, cash, skip, found, indicate, conclusion, identify, coconut, value, mom, mild, clear, historical, girl, manufacture, create, viral, accusation, opposition, male, pork, terminate, slim, database, repeat, legislation, post, parliament, draft, primary, robot, announce, government
Added Edges	(found, area), (found, historical), (found, value), (found, party), (found, welfare), (found, government), (government, announce), (found, cash), (government, accusation), (government, party), (viral, information), (government, parliament), (legislation, terminate),
Overlapped Nodes	cannabis, baht, area, year, Thai, news, oil, human, son, channel, do, expose, government, Bangkok, support, people
Overlapped Edges	(Thai, people), (channel, Bangkok), (Thai, son), (news, son), (year, Bangkok), (year, government), (news, government), (support, area), (baht, do), (channel, people), (support, expose), (channel, government), (support, do), (news, expose), (year, son), (support, people), (government, Bangkok), (support, cannabis), (support, oil)

D. Building Graph Evolutionary Properties

Considering a time-evolved analysis in a certain period, the graph evolutionary assists the interpretation of expanded connections when the time changes. Since a simplified graph is representative of daily tweet messages, we compose an evolution of graph-based representation into two approaches; added and overlapped nodes/edges. The nodes/edges, which join on the next day (d_{i+1}), but do not appear in d_i is called "added nodes/edges". However, the "overlapped nodes/edges" are the nodes/edges that exist in both d_i and d_{i+1}. An example of graph evolutionary properties including added and overlapped nodes/edges are shown in Table I where retrieved tweet messages from two popular news agencies in Thailand on 14–15 June, 2022.

IV. EXPERIMENTAL DESIGN

A. Dataset

This study pulled out tweet messages from Thai news agencies through Twitter API including two accounts: MorningNewsTV3 and Matichon Online. Both are the most renowned news industries in Thailand which have over 3.1 and 1.1 million followers respectively. The number of collected data from both accounts in June-July 2022 is displayed in Table II. However, a combination of two famous agencies is used in the experiment. All tweet messages are applied to the data preprocessing and tokenization step.

B. Construct a Simplified Network

After the data preparation is completed, we can construct a network graph in which vertices represent tokens and edges represent a link between adjacent tokens. Since we desire to keep only important vertices/edges, the less important vertices/edges will be disconnected from the original network. Hence, the original network is simplified by selecting only the top 30 vertices of DC, BC, CC, and CF. We sample tweet messages for 14 days in the experiment which repeat all previous steps accordingly.

TABLE II: TOTAL NUMBER OF COLLECTED TWEETS

	Matichon Online	Morning News TV3
1-7 June 2022	612	464
8-14 June 2022	704	551
15-21 June 2022	1,015	632
22-30 June 2022	1,489	692

C. Evaluation Metrics

To assess the experiment result in this study, the performance of a proposed framework is evaluated by precision, recall, and F-score. Precision measures the number of relevant samples out of the total number of relevant predictions. However, a recall is a fraction of correctly relevant predictions and the total number of actual relevant samples. F-score is the harmonic mean between precision and recall. Therefore, these indicators are used for the performance evaluation between our framework and baseline models.

TABLE III: EXPERIMENTAL CONFIGURATIONS

Parameters	Configurations
Selected top N vertices (Simplified graph)	30
Selected top N vertices (Pruned neighborhoods)	5
Number of topics used in topic models	10
Similarity threshold	0.85

V. EVALUATION AND RESULTS

A. Retrieved Related Tweets Approach

This section discussed the method of retrieving related tweets and provides the performance comparison among baseline approaches in the latter. Searching for any information on the internet, we typically input a query that we would like to find. Thus, a query term is required as a reference input before retrieving the information.

To discover tweets related to a query term, different approaches use diverse retrieval methods. After a collection of tweets have retrieved, human-annotated judgment is essential in this study to examine the relevant tweets. Three annotators considered each retrieved tweet message, and their opinions are operated using a majority vote to finalize the result.

In our framework, we initially take a query term to find the edges in the simplified graph. Within 2-hop neighborhoods connected to the vertex are considered as a set of possible edges. Furthermore, we extend a pruning process to aid in keeping only relevant 2-hop vertices by utilizing the top five vertices of DC and CF. Then, the final set of edges is used to find word co-occurrence in the sample tweets.

The first baseline approach, Word Mover's Distance (WMD), is a novel approach that measures the distance that the embedded words of one document require to travel to the embedded words of another document [30]. The embedded query term measures a similarity among the embedded tweet messages. The WMD similarity is implemented using the Gensim Python library which provides a useful library to compute the similarity between documents. Tweets that have a similarity score greater than a given threshold will be considered the retrieved tweets.

The second baseline approach called Soft Cosine Measure (SCM) is a state-of-the-art approach to the semantic text similarity task. SCM assesses a similarity between words using word2vec embeddings [31]. Also, the SCM similarity utilizes the Gensim library to compute a score regarding the query term. Only similarity scores greater than a given threshold will be considered in the retrieved tweets.

Latent Dirichlet Allocation, also known as LDA, is a generative probabilistic model of a text corpus that has been widely studied in diverse areas of natural language processing [32]. LDA is one of the most renowned techniques in topic modeling which facilitates semantic mining and topic discovery among documents. Using cosine similarity to compare a query term and model topics derives a similarity score. Also, a given threshold is defined as a relevant tweet if a similarity score is higher.

TABLE IV: PERFORMANCE EXPERIMENT

Topic No.	Metrics	WMD	SCM	LDA	Our framework
1	Precision	0.48919	0.54304	0.5279	0.99536
	Recall	0.5	0.83535	0.60341	0.75423
	F-score	0.49453	0.52698	0.53294	0.83475
2	Precision	0.50041	0.51564	0.54819	0.90242
	Recall	0.50073	0.66892	0.59732	0.93514
	F-score	0.49916	0.30753	0.56206	0.91811
3	Precision	0.49358	0.50201	0.5065	0.51432
	Recall	0.5	0.51656	0.5482	0.61082
	F-score	0.49677	0.11922	0.49191	0.50285
4	Precision	0.55218	0.51466	0.51975	0.63529
	Recall	0.51989	0.66396	0.60612	0.77829
	F-score	0.52776	0.48295	0.52036	0.68022
5	Precision	0.49367	0.50339	0.50026	0.84116
	Recall	0.5	0.54896	0.50136	0.60653
	F-score	0.49681	0.44813	0.49037	0.66082
6	Precision	0.65879	0.5301	0.5179	0.61858
	Recall	0.50757	0.57057	0.52888	0.62419
	F-score	0.49477	0.28426	0.51732	0.62129
7	Precision	0.54378	0.52918	0.5142	0.66545
	Recall	0.57539	0.56432	0.52867	0.91237
	F-score	0.55189	0.16913	0.51289	0.71907
8	Precision	0.48991	0.51024	0.54985	0.73692
	Recall	0.49943	0.5945	0.72727	0.82098
	F-score	0.49463	0.22009	0.56533	0.77223
9	Precision	0.53286	0.53376	0.51591	0.83205
	Recall	0.54226	0.5444	0.51944	0.83515
	F-score	0.53577	0.19889	0.51679	0.83359
10	Precision	0.50103	0.51986	0.5096	0.88013
	Recall	0.50508	0.78947	0.53558	0.89423
	F-score	0.48941	0.43507	0.50783	0.88705

B. Experimental Results

To assess the performance in the retrieval of relevant documents, we download tweets messages from two popular Thai news agency accounts in June–July, 2022. A 14-day window size is used for tracing the propagation of news topics. In this experiment, we evaluate the performance measured by precision, recall, and F-score from 10 news trending topics. A set of parameter configurations in the experiment are given in Table III.

The performance from 10 selected topics in the experiment shows that our framework significantly outperformed baseline models on all metrics as shown in Table IV. To compare among baseline approaches, SCM was a leading of the average recall, and WMD slightly performed better in terms of the average precision. LDA method showed the most F-score on average. However, the average F-score of our framework gained 29.78% compared to the LDA model. Also, the average precision and recall were increased by 31% and 19% respectively.

With the benefits of graph evolution, our framework can widely retrieve tweet messages where the situations have evolved to the others that related to a similar topic. On the other hand, using a similarity score of the word vector embedded approaches might not find relevant tweets when the situations have changed. Furthermore, simplifying and pruning processes assist to preserve important information and increase the recall score.

VI. CONCLUSION

In this paper, we explored the benefits of an evolutionary in graph-based representation for obtaining relevant tweet messages. With the use of a simplification-based technique in graph-based summarization, our framework maintains important information in the network based on centrality measurement and clustering coefficient. Meanwhile, the diffusion of relations is significant to find more relevant information using graph evolution. In the experiment, the proposed framework showed an outstanding performance against word-embedded similarity methods. When the events have developed to the others, using word-embedded vector approaches can only find related tweets on the current situation, but the expanded situations corresponding to the same topic might not be extensively retrieved. In our approach, not only the current situations will be considered as related tweets, but more relevant tweet messages with time-evolved situations will also be retrieved which significantly improved the recall as shown in the experimental results.

Conflict of Interest

The authors hereby declare that they have no conflict of interest.

Author Contributions

Patta Yovithaya developed the framework, conducted the experiments, and wrote a first version of the paper. Sukree Sinthupinyo supervised the research and revised the paper. All authors had approved the final version of manuscript.

References

[1] D. Suryadi, “The potential of emotions as predictors of news popularity on twitter,” in *Proc. 2021 International Conference on Data Analytics for Business and Industry (ICDABI)*, 2021.

[2] B. Wang, B. Liu, and Q. Zhang, “An empirical study on Twitter’s use and crisis retweeting dynamics amid Covid-19,” *Natural Hazards*, vol. 107, no. 3, pp. 2319–2336, 2021.

[3] F. Harary and G. Gupta, “Dynamic graph models,” *Mathematical and Computer Modelling*, vol. 25, no. 7, pp. 79–87, 1997.

[4] D. L. Hansen *et al.*, “Chapter 3 — Social network analysis: Measuring, mapping, and modeling collections of connections,” in *Analyzing Social Media Networks with NodeXL (Second Edition)*, Morgan Kaufmann, 2020, pp. 31–51.

[5] R. Mihalcea and P. Tarau, “Textrank: Bringing order into texts,” Association for Computational Linguistics, EECS News, 2004.

[6] C. Yongkiatpanich and D. Wichadakul, “Extractive text summarization using ontology and graph-based method,” in *Proc. 2019 IEEE 4th International Conference on Computer and Communication Systems (ICCCS)*, 2019.

[7] M. L. Joshi, N. Joshi, and N. Mittal, “SGATS: Semantic graph-based automatic text summarization from hindi text documents,” *ACM Trans. Asian Low-Resour. Lang. Inf. Process*, vol. 20, no. 6, article 102, 2021.

[8] A. A. Natesh, S. T. Balekuttira, and A. P. Patil, “Graph based approach for automatic text summarization,” *International Journal of Advanced Research in Computer and Communication Engineering*, vol. 5, no. 2, pp. 6–9, 2016.

[9] A. Khan *et al.*, “Abstractive text summarization based on improved semantic graph approach,” *International Journal of Parallel Programming*, vol. 46, no. 5, pp. 992–1016, 2018.

[10] W. Li *et al.*, “Leveraging graph to improve abstractive multi-document summarization,” arXiv preprint arXiv:2005.10043, 2020.

[11] J. Tan, X. Wan, and J. Xiao, “Abstractive document summarization with a graph-based attentional neural model,” in *Proc. the 55th Annual Meeting of the Association for Computational Linguistics*, 2017.

[12] T. Uçkan and A. Karcı, “Extractive multi-document text summarization based on graph independent sets,” *Egyptian Informatics Journal*, vol. 21, no. 3, pp. 145–157, 2020.

[13] R. C. Belwal, S. Rai, and A. Gupta, “A new graph-based extractive text summarization using keywords or topic modeling,” *Journal of Ambient Intelligence and Humanized Computing*, vol. 12, no. 10, pp. 8975–8990, 2021.

[14] W. S. El-Kassas *et al.*, “EdgeSumm: Graph-based framework for automatic text summarization,” *Information Processing & Management*, vol. 57, no. 6, 102264, 2020.

[15] S. Ullah and A. A. A. Islam, “A framework for extractive text summarization using semantic graph based approach,” in *Proc. the 6th International Conference on Networking, Systems and Security*, 2019.

[16] Y. Liu *et al.*, “Graph summarization methods and applications: A survey,” *ACM Comput. Surv.*, vol. 51, no. 3, article 62, 2018.

[17] S. Zeqian, M. Kwan-Liu, and T. Eliassi-Rad, “Visual analysis of large heterogeneous social networks by semantic and structural abstraction,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 12, no. 6, pp. 1427–1439, 2006.

[18] C. Li and S. Lin, “Egocentric information abstraction for heterogeneous social networks,” in *Proc. 2009 International Conference on Advances in Social Network Analysis and Mining*, 2009.

[19] D. Hennessey *et al.*, “A simplification algorithm for visualizing the structure of complex graphs,” in *Proc. 2008 12th International Conference Information Visualisation*, 2008.

[20] Y. Li, Q. Zhang, and T. Reps, “Fast graph simplification for interleaved Dyck-reachability,” in *Proc. the 41st ACM SIGPLAN Conference on Programming Language Design and Implementation*, 2020, Association for Computing Machinery: London, UK. pp. 780–793.

[21] N. Ruan, R. Jin, and Y. Huang, “Distance preserving graph simplification,” in *Proc. 2011 IEEE 11th International Conference on Data Mining*, 2011.

[22] N. Panagiotou, I. Katakis, and D. Gunopulos, “Detecting events in online social networks: Definitions, trends and challenges,” in *Solving Large Scale Learning Tasks. Challenges and Algorithms*, 2016, pp. 42–84.

[23] M. Cordeiro and J. Gama, “Online social networks event detection: A survey, in solving large scale learning tasks,” in *Challenges and Algorithms*, Springer, 2016, pp. 1–41.

[24] W. Dou *et al*., “Event detection in Social media data,” in *Proc. IEEE VisWeek Workshop on Interactive Visual Text Analytics-Task Driven Analytics of Social Media Content*, 2012.

[25] S. Srinivasan *et al*., “Chapter three — Machine learning techniques for fractured media,” in *Advances in Geophysics*, B. Moseley and L. Krischer, Eds. Elsevier, 2020, pp. 109–150.

[26] U. Brandes, “On variants of shortest-path betweenness centrality and their generic computation,” *Social Networks*, vol. 30, no. 2, pp. 136–145, 2008.

[27] C. Perez and R. Germon, “Chapter 7 — Graph creation and analysis for linking actors: Application to social data,” in *Automating Open Source Intelligence*, R. Layton and P. A. Watters, Eds. Syngress: Boston, 2016, pp. 103–129.

[28] J. Saramäki *et al.*, “Generalizations of the clustering coefficient to weighted complex networks,” *Physical Review E.*, vol. 75, no. 2, 027105, 2007.

[29] Twitter API. [Online]. Available: https://developer.twitter.com/en/docs/twitter-api

[30] M. J. Kusner *et al.*, “From word embeddings to document distances,” in *Proc. the 32nd International Conference on International Conference on Machine Learning*, JMLR.org: Lille, France, 2015, pp. 957–966.

[31] G. Sidorov *et al*., “Soft similarity and soft cosine measure: Similarity of features in vector space model,” *Computación y Sistemas*, vol. 18, no. 3, pp. 491–504, 2014.

[32] H. Jelodar *et al.*, “Latent Dirichlet Allocation (LDA) and topic modeling: Models, applications, a survey,” *Multimedia Tools Appl.*, vol. 78, no. 11, pp. 15169–15211, 2019.

Copyright © 2023 by the authors. This is an open access article distributed under the Creative Commons Attribution License which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited (CC BY 4.0).