Graph-Based Clustering of Short Texts Using Word Embedding Similarity

Supakpong Jinarat, Ratchakoon Pruengkarn
Information Technology and Artificial Intelligence Program,
College of Engineering and Technology, Dhurakij Pundit University, Bangkok, Thailand 10210

Abstract—**The exponential growth of short textual content on the Internet, such as social media posts and search snippets, necessitates effective text mining techniques. Short text clustering, a critical tool for organizing this data, contends with two primary challenges: data sparsity, which undermines the quality of traditional clustering methods, and the poor interpretability of machine-generated cluster labels. This study introduces the Semantic Word Graph (SWG) algorithm, a novel graph-based approach designed to address both of these issues simultaneously. Our methodology begins by constructing a global word graph where nodes represent unique terms from the corpus, and edges are weighted by the semantic similarity of word pairs, calculated using a pre-trained Word2Vec model. Cohesive communities of words are then identified using the Louvain method, and documents are assigned to clusters based on these communities. Meaningful cluster labels are generated by ranking representative nouns within each community. To validate our approach, the SWG algorithm was evaluated on three benchmark datasets (AG News, Tweet, and SearchSnippets) and compared against established methods, including Lingo, Suffix Tree Clustering (STC), and K-means. Quantitative results, measured by the F-score, show that SWG achieved up to 0.89 F-score on AG News, 0.85 on Tweets, and 0.82 on SearchSnippets, consistently outperforming baseline algorithms in clustering quality. Furthermore, a qualitative analysis confirms that SWG produces more coherent and topically comprehensive cluster labels, improving interpretability. This study concludes that the SWG algorithm is a robust and effective framework for enhancing both the accuracy and interpretability of short text clustering. Future research could explore integrating contextual embeddings such as BERT to capture deeper semantic relationships, optimizing the similarity threshold dynamically for different datasets, and scaling the algorithm to handle larger, real-time streaming text data. These directions would further improve the applicability of SWG in diverse domains such as social media analytics, news aggregation, and real-time topic detection.**

Keywords*—*Clustering; graph-based clustering; semantic similarity; short text; word embedding

I. Introduction

Short text clustering is a specialized form of text clustering that focuses on grouping brief textual content—such as tweets, comments, and news headlines—into coherent categories based on their content. Unlike traditional text clustering methods that deal with longer documents, short text clustering faces unique challenges due to data sparsity, informal language usage, and high noise levels. With the exponential growth of short text data on social media platforms, news feeds, and online forums, the demand for accurate and interpretable clustering methods has become increasingly critical.

Current approaches often suffer from degraded clustering quality when dealing with sparse data and have difficulty generating labels that are both meaningful and representative of the underlying topics. These limitations reduce the practical usability of clustering results in real-world applications such as trend detection, news aggregation, and customer feedback analysis. Moreover, while deep learning and embedding-based models have improved text representation, many methods still fail to fully exploit semantic relationships between words in short texts, leading to clusters that lack coherence.

Although several clustering techniques, including graph-based and embedding-enhanced methods, have been proposed, most still struggle with balancing clustering accuracy and label interpretability. Few approaches explicitly address both challenges in a unified framework, and existing solutions often require extensive parameter tuning or rely on domain-specific data, limiting their generalizability.

This study aims to address these challenges by introducing the Semantic Word Graph (SWG) algorithm, a novel graph-based clustering approach designed specifically for short text. The proposed method enhances data representation using pre-trained word embeddings, detects cohesive semantic communities, and generates interpretable cluster labels through part-of-speech-based ranking. By focusing on both clustering quality and label interpretability, this research seeks to deliver a more robust and practical solution for real-world short text analysis.

II. Related Work

Despite the challenges, short text clustering has a wide range of real-world applications. In social media analysis, it is used to identify trending topics [1], [2], [3], [4], [5], track public opinion [6], and detect events [7]. In news aggregation, it helps organize news into structured categories. In customer service, clustering improves response time by categorizing customer inquiries. It is also applied in sentiment analysis to classify short texts into positive, negative, or neutral categories [8], [9].

Recently, neural network-based approaches have been adopted for short text representation learning. For example, [10] proposed a distributional semantic model to improve semantic understanding in clustering, while [11] developed STC2, which applies convolutional neural networks (CNN) [12] to feature learning. STC2 converts short text features into compact binary codes and applies k-means for clustering.

Word embedding is a procedure used in Natural Language Processing (NLP) and machine learning to transform individual words to the numerical value of vector. Word embedding enables computers to analyze and process human language by transforming words into numerical vectors which reveal some semantic relationships between them.

Word2vec is a specific algorithm to learn word embeddings from large text corpus. It was developed by a team of researchers at Google in 2013 [13]. The algorithm employs a neural network to acquire the contextual associations among words within an extensive text corpus. Neural network is trained to predict a word appearing given its surrounding words. The weights of the neural network are used as the word embeddings.

Word embeddings are powerful because they capture the relationships between words in a language. For example, in a vector space created by word embeddings, words that are semantically similar are located close to each other. This allows computers to perform semantic tasks such as finding synonyms, detecting word relationships, and even predicting the sentiment of text.

The fundamental concept underlying word2vec involves utilizing a neural network to forecast the context words that encompass a target word, leveraging a substantial text corpus. The neural network is trained to encode each word as a vector of real values, ensuring that similar words are positioned close together in the high-dimensional space. This is achieved through a training process called stochastic gradient descent, which updates the weights of the neural network in a way that minimizes the difference between the predicted and actual context words.

There are two primary architectures of word2vec. First, Skip-gram model is to predict context words by using a target word. Secondly, Continuous Bag-of-Words (CBOW) model is to predict the target word based on its context. Both architecture have demonstrated their effectiveness in generating high-quality word embeddings, and the selection between them relies on the task at hand and the corpus size.

III. Methodology

In this research, we introduce a clustering algorithm designed specifically for short text documents. Our algorithm includes six primary steps, as shown in Fig. 1. These steps include pre-processing, construction of a word semantic graph, subgraph detection, document assignment, cluster selection, and cluster label generation. By following this methodology, our aim is to produce high-quality clusters with meaningful results.

A. Preprocessing

Pre-processing is an essential step in text clustering to clean and transform raw text data into a suitable format for clustering algorithms. We remove special characters, punctuation marks, and numbers from the text then eliminate stop words (commonly used words that do not carry significant meaning) such as ‘and’, ‘the’, ‘is’, etc. Irrelevant or noisy elements such as URLs, HTML tags, and email addresses are removed. For Tweet dataset, Twitter-specific elements like hashtags, mentions, and retweet are removed as well.

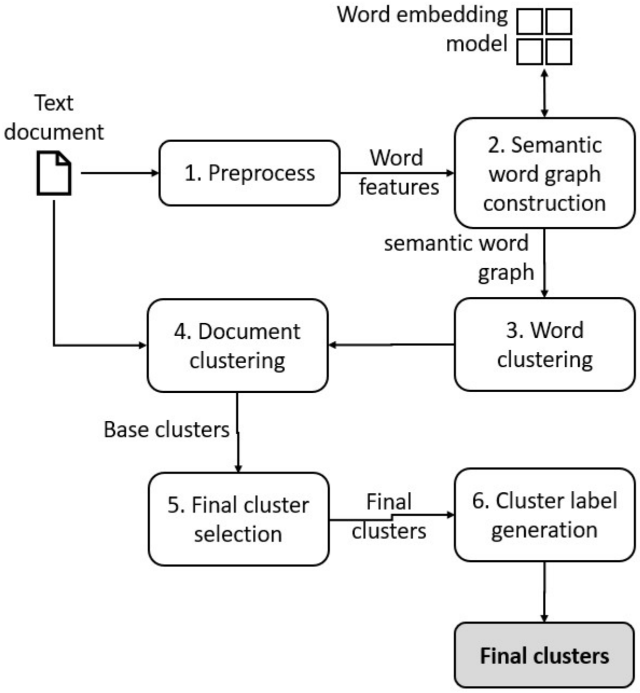

Fig. 1. System architecture of our proposed algorithm.

B. Semantic Word Graph Construction

After preprocessing, a semantic word graph is constructed based on the word features obtained from the previous step. We utilize a pre-trained Word2Vec model trained on the Google News corpus, containing approximately 100 billion words. The model provides 300-dimensional vector representations for approximately three million unique words [14].

The graph is defined as $G = (V, E, W)$, where V is the set of unique words from the preprocessed text. E is the set of undirected edges e_{ij} between word pairs v_i , $v_j \in V$ and W is the set of edge weights w_{ij}, which represent the semantic similarity between words v_i and v_j computed using the cosine similarity of their word vectors.

$$w_{ij} = sim(v_i, v_j) = \frac{v_i \cdot v_j}{\|v_i\|\|v_j\|} \tag{1}$$

Only edges with weights exceeding a predefined threshold (e.g. $w_{ij} > 0.5$) are retained to construct a meaningful semantic network.

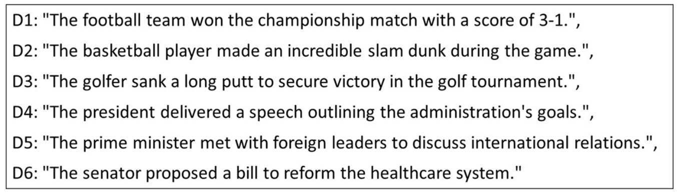
D1: "The football team won the championship match with a score of 3-1.",
D2: "The basketball player made an incredible slam dunk during the game.",
D3: "The golfer sank a long putt to secure victory in the golf tournament.",
D4: "The president delivered a speech outlining the administration's goals.",
D5: "The prime minister met with foreign leaders to discuss international relations.",
D6: "The senator proposed a bill to reform the healthcare system."

Fig. 2. Example of text documents used for demonstration of word semantic graph construction.

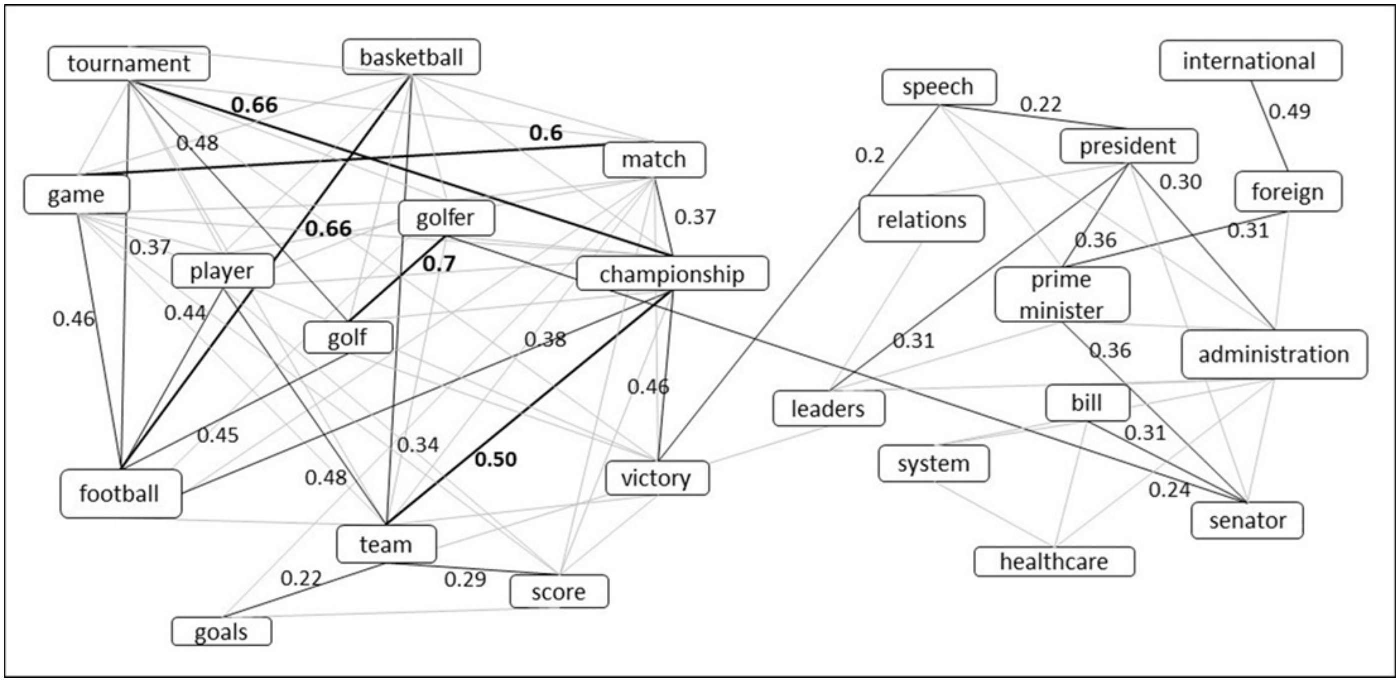

Fig. 3. Illustrative of word semantic graph construction from text document example.

Fig. 3 showcases the construction of the word semantic graph, derived from the example text document illustrated in Fig. 2.

C. Word Clustering

In the semantic word graph produced from the previous step, words with high semantic similarity are connected by edges with weights close to 1. To identify cohesive groups of related words, the graph is divided into subgraphs (or word clusters) by pruning weak edges—those with weights below a predefined similarity threshold st.

After pruning, we apply the Louvain method [15] for community detection to each resulting subgraph. This algorithm detects communities based on modularity optimization, helping identify groups of words that exhibit strong semantic relationships. As shown in Fig. 4, when a similarity threshold of 0.3 is applied, the semantic word graph is partitioned into two distinct subgraphs. Each subgraph contains strongly connected words, which are treated as word clusters. These word clusters serve as the input for the next step: document clustering, where documents are grouped based on their semantic similar derived from these clusters.

Each subgraph produced from this step is treated as a word cluster, with each cluster containing a set of candidate labels—i.e., representative words from the cluster—but no associated documents at this stage. The assignment of documents to these clusters will be operated in the next step of the algorithm.

As illustrated in Fig. 4, two example word clusters are presented.

The first word cluster includes candidate labels such as: “football”, “team”, “championship”, “match”, “score”, “basketball”, “player”, “game”, “tournament”, “golfer”, “victory”, “golf”, and “goals”.

The second word cluster contains labels like: “president”, “administration”, “prime minister”, “foreign”, “leaders”, “international”, “senator”, and “bill”.

These clusters are semantically coherent and form the basis for the upcoming step of document assignment based on semantic matching.

D. Document Clustering

The semantic word subgraphs produced from the previous phase are used as word clusters for the purpose of document clustering. Each document is assigned to one or more clusters based on the presence of any word from the set of candidate labels within the document text. This approach allows a document to be associated with multiple word clusters, resulting in overlapping clustering. Such flexibility is useful in cases where a document covers multiple topics or contains ambiguous content. The resulting groups of documents—each associated with a semantic word cluster—are referred to as base clusters, which used as input for the next stages of the clustering framework.

E. Final Cluster Selection

In the previous step, we define the importance of a base cluster based on its size—that is, the number of documents it contains. Accordingly, we rank all base clusters in descending order of size. The top k clusters in this ranking are selected as the final output clusters for further interpretation and evaluation.

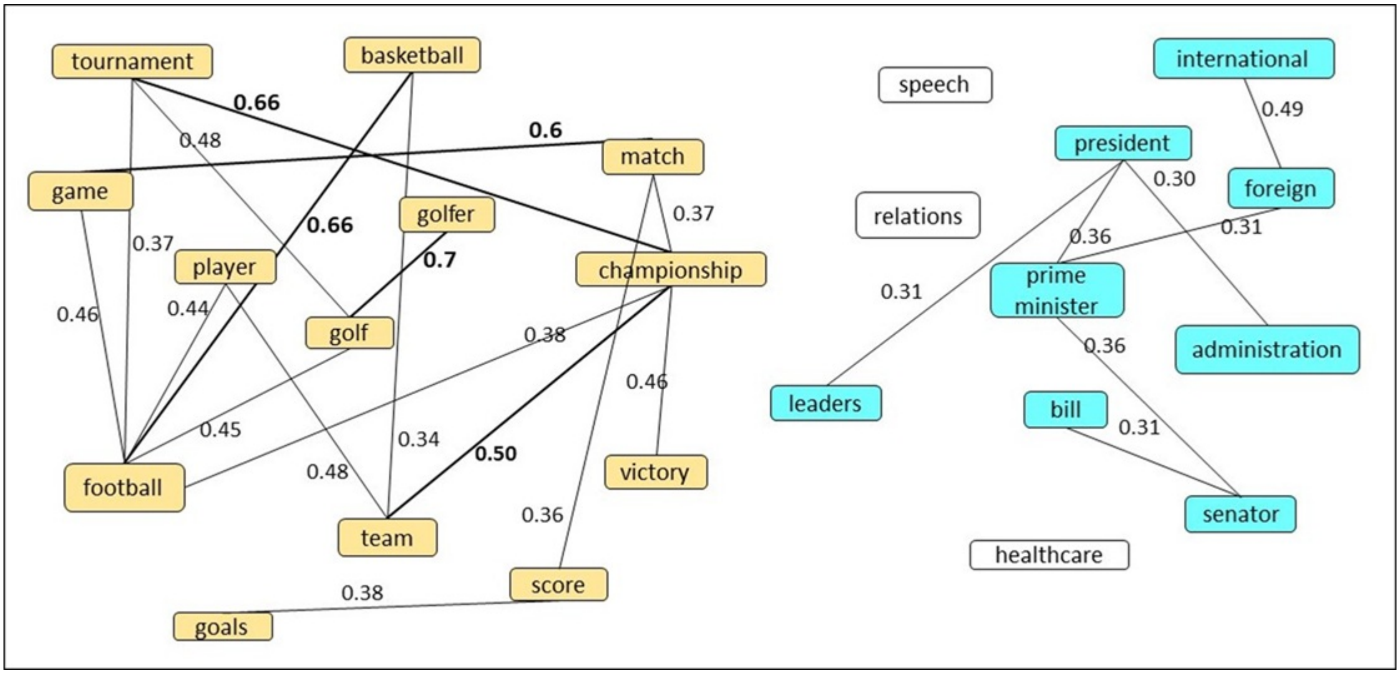

Fig. 4. An illustrative instance of a word semantic graph and its corresponding subgraphs is presented, utilizing a similarity threshold of 0.3. The semantic word graph displays the interconnections among words, while the subgraphs depict clusters formed based on the established similarity threshold.

F. Generating Cluster Label

Beyond clustering efficiency, it is equally essential to provide a descriptive interpretation of each resulting cluster. In this step, we focus on generating cluster labels for the final output clusters. As each semantic subgraph represents a set of semantically related words, we consider these words as candidate descriptors for the associated documents. Specifically, we first eliminate non-noun words using part-of-speech filtering. Next, each word is assigned a relevance score computed using a scoring function [see Eq. (2)], which quantifies its representativeness.

$$label\_score(f_i) = df(f_i) \times neighbor(v_i) \tag{2}$$

Where $df(f_i)$ is document frequency of word f_i from the input text document and neighbor(vi) is number of nodes that links to the node v_i in the semantic word graph.

Finally, the top-ranking words—typically those with the highest scores—are selected as cluster labels, offering an interpretable summary of each semantic group.

IV. Experiments

A. Experimental Setup

We evaluated the performance of our proposed algorithm, Semantic Word Graph (SWG), by comparing it with established text clustering methods, including Lingo [16], Suffix Tree Clustering (STC) [17], [18] and the K-means clustering algorithm. For both Lingo and STC, default parameter settings recommended by their original implementations were adopted.

For SWG, we experimented with three different values of the similarity threshold: $st = 0.4$, $st = 0.5$, and $st = 0.6$. The performance of all algorithms was assessed across a range of cluster counts, from 1 to 10, which aligns with the actual number of topics in each dataset used.

B. Dataset

The experiments were conducted on three widely used benchmark datasets for short text clustering: namely the SearchSnippets, Tweet, and AG News datasets.

For the SearchSnippets dataset, which was originally compiled by [19], we used a collection of 12,340 short text snippets retrieved from web search results. These snippets were obtained using predefined keywords associated with 8 distinct topical categories. A summary of this dataset is shown in Table I.

TABLE I. The 8 Different Topics SearchSnippets Dataset

Topic	Number of Documents
Business	1,500
Computers	1,500
Culture-Arts-Entertainment	2,210
Education-Science	2,660
Engineering	370
Health	1,180
Politics-Society	1,500
Sports	1,420

Secondly, for the Tweet dataset, we collected tweets via web scraping from 16 prominent public Twitter accounts, focusing on 4 specific topics, as summarized in Table II. A maximum of 2,000 tweets was retrieved from each account on 14 May 2022. The retrieved tweets were categorized according to their corresponding topics. Subsequently, 2,000 tweets were randomly selected for each topic, resulting in a final dataset of 8,000 tweets (across four topics) used in our experimental analysis

TABLE II. The Twitter Accounts were Categorized into 4 Distinct Tweet Topics, Ensuring Separation based on their Content

Politics	Sports	Sci/Tech	Business
BBC_Politics	ESPN	NASA	CNNBusiness
CNN_Politics	Fox_Sports	Science_News	Davos
POLITICO	SportsCenter	science	economics
Post_Politics	Twitter_Sports	CERN	EconomicTimes

Lastly, we used the AG News dataset, a large corpus of online news articles commonly employed in text classification and clustering tasks [20]. This dataset contains 496,835 news articles collected from over 2,000 distinct news outlets. For our experiments, we randomly selected 1,000 articles from each six categories, resulting in a total of 6,000 documents. Only the title and description fields were used for clustering. A summary of this dataset is presented in Table III.

TABLE III. AG News Dataset Separated by 6 Different Topics

Topic	Number of Documents
Business	1,000
Entertainment	1,000
Health	1,000
Sci/Tech	1,000
Software and Development	1,000
Sports	1,000

C. Evaluation Method

We evaluated the effectiveness of our proposed clustering algorithm by measuring its ability to accurately group short texts and address the inherent challenges of short-text data. Experiments were conducted on multiple benchmark datasets consisting of short text documents. Our method was compared with established clustering techniques, including Lingo and Suffix Tree Clustering (STC). The primary evaluation metric used was the F-score.

The F-score is a widely used evaluation metric in machine learning, particularly for classification and clustering tasks. It represents the harmonic mean of precision and recall, thereby providing a balanced measure of accuracy and completeness.

Precision is the proportion of true positive predictions among all positive predictions made by the model. It quantifies the accuracy of positive assignments.

$$\text{Precision} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}} \tag{3}$$

Recall is the proportion of true positive predictions among all actual positive instances. It reflects the completeness of the positive predictions.

$$\text{Recall} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}} \tag{4}$$

The F-score can be computed using two standard variants: micro-averaging and macro-averaging. The distinction between these methods lies in how precision and recall are aggregated across multiple clusters.

The micro-average F-score [21], [22], [23], [24] computes performance globally by counting the total true positives, false positives, and false negatives across all clusters. It treats the entire clustering task as a single binary classification problem, where documents that belong to any cluster are considered the positive class. This approach is particularly useful when class imbalance exists, as it emphasizes frequent categories. The micro-average F-score is computed using the following formula:

$$F_{\text{micro}} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{5}$$

In contrast, the macro-average F-score computes the F-score for each cluster individually and then averages the results across all clusters. In this case, each cluster is treated as a separate binary classification task, allowing equal weight to be assigned to both large and small clusters. The macro-average F-score is calculated using the following formula:

$$F_i = 2 \times \frac{\text{Precision}_i \times \text{Recall}_i}{\text{Precision}_i + \text{Recall}_i} \tag{5}$$

where Precisioni and Recalli are computed for the i-th cluster. The F-score macro-average [25], [26] is then computed as the average of the F-scores for all clusters:

$$F_{\text{macro}} = \frac{\sum F_i}{n} \tag{7}$$

where n is the total number of clusters.

Moreover, we propose a modification to the traditional macro-average F-score, referred to as the weighted macro-average F-score. This metric considers the size of each cluster when computing the average, thus giving more weight to larger clusters. The weighted macro-average F-score is defined as follows:

$$F_{\text{weighted\_macro}} = \frac{\sum (F_i \times n_i)}{N} \tag{8}$$

where n is the total number of documents, F_i denotes the F-score of the $i - th$ cluster, and n_i represents the number of documents in cluster. The weighted macro-average F-score offers a more representative evaluation of clustering performance, as it accounts for the varying sizes of clusters and emphasizes the contribution of each cluster to the overall clustering quality.

V. Result and Discussion

A. The Quality of Clustering

To ensure a fair comparison among Lingo, STC, and our proposed method (SWG), we evaluated all algorithms based on the top-k clusters, where k ranges from 2 to 10. This aligns with the actual number of clusters present in each dataset used in the experiments.

The results clearly show that the proposed SWG algorithm consistently outperformed the baseline methods, particularly on the Tweet dataset, as illustrated in Fig. 6. These findings highlight the significant advantage of incorporating semantic word graphs, especially when dealing with short texts that contain sparse and noisy information.

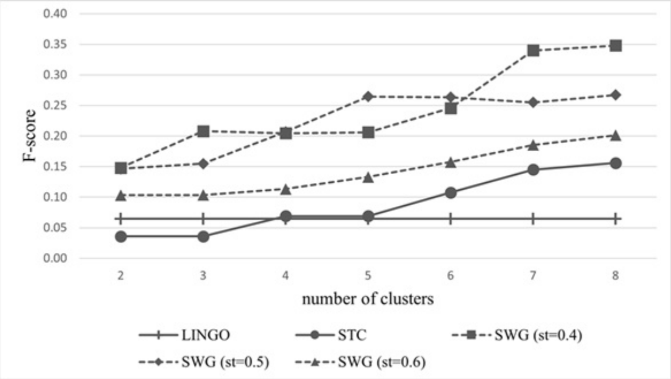

Fig. 5. Quality of clustering on the AG news dataset.

For the AG News dataset (Fig. 5), our method achieved the highest F-score across all tested similarity thresholds, demonstrating its effectiveness in grouping short news articles based on semantic similarity.

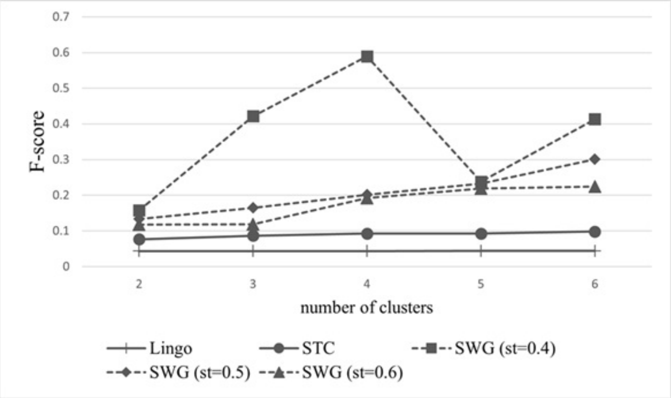

Fig. 6. Quality of clustering on the tweet dataset.

Likewise, on the Tweet dataset, SWG achieved the best F-score under all evaluated threshold values, confirming its robustness in handling short, informal, and noisy text typical of social media content. These results underscore the practical advantage of semantic-based clustering in real-world applications involving brief and informal textual data.

As illustrated in Fig. 7, our proposed method achieved higher F-scores than the baseline algorithms at two similarity threshold values ($st = 0.5$ and $st = 0.6$). However, the result for the remaining threshold ($st = 0.4$) was slightly lower than those obtained by the comparative methods.

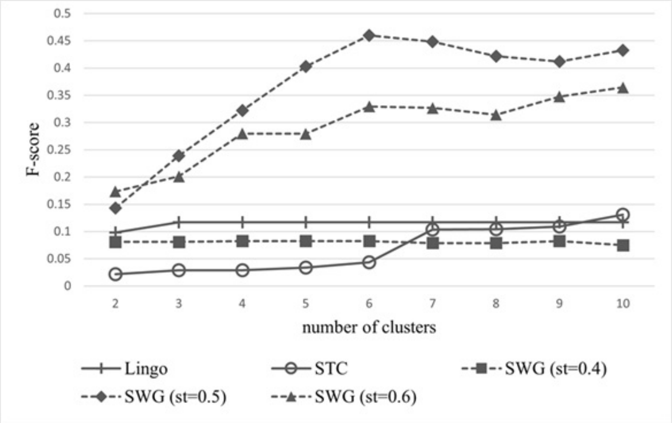

Fig. 7. Quality of clustering on the search snippets dataset.

Despite this, our method still demonstrated competitive performance in clustering short texts from search result snippets, which are often fragmentary and challenging to group semantically. These results further support the robustness of the proposed algorithm across different datasets and parameter settings.

In this study, we also compared the performance of our graph-based text clustering algorithm (SWG) with the well-known vector-based baseline, K-means.

As shown in Fig. 8 and Fig. 10, the SWG algorithm consistently outperformed K-means across all tested cluster numbers on both the AG News and SearchSnippets datasets.

For the Tweet dataset (Fig. 9), our algorithm exhibited slightly lower performance than K-means when the number of clusters was set to 2 and 3. However, SWG outperformed K-means for cluster numbers 4, 5, and 6—most notably when $k = 4$, which corresponds to the ground truth number of clusters in the Tweet dataset. This result demonstrates the strength of SWG in capturing semantically coherent clusters when aligned with real-world topic distributions.

B. Representative Words for a Cluster Result

In this section, we analyze the representative words generated by the three clustering algorithms—Lingo, STC, and SWG—across all three datasets. These representative words reflect the dominant themes within each cluster and allow for qualitative comparisons between the algorithms.

Table IV presents the representative words produced by each algorithm on the AG News dataset, which includes six primary categories: Business, Entertainment, Health, Science/Technology, Software and Development and Sports.

The Lingo algorithm captured Sports-related topics in Cluster 2, 4, and 6 with words like Wins, Wins Gold, and Phelps Wins. It also identified Health-related topics in Cluster 1 and 5 with terms such as Drug and Cancer Drug.

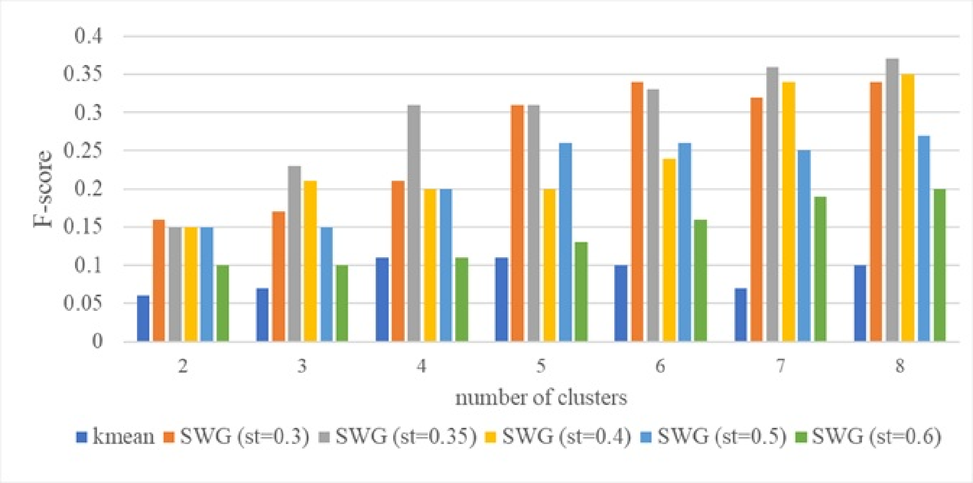

Fig. 8. Quality of clustering of our algorithm compared with k-means algorithm on AG news dataset.

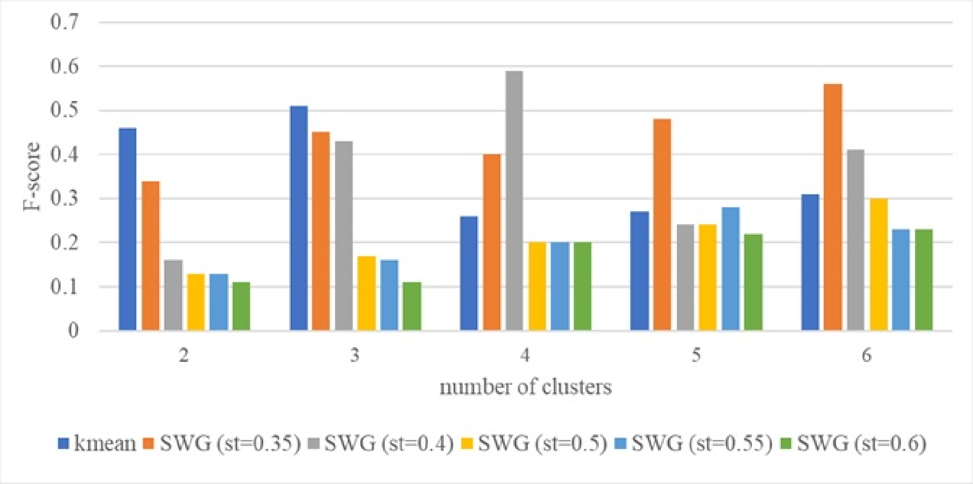

Fig. 9. Quality of clustering of our algorithm compared with k-means algorithm on tweet dataset.

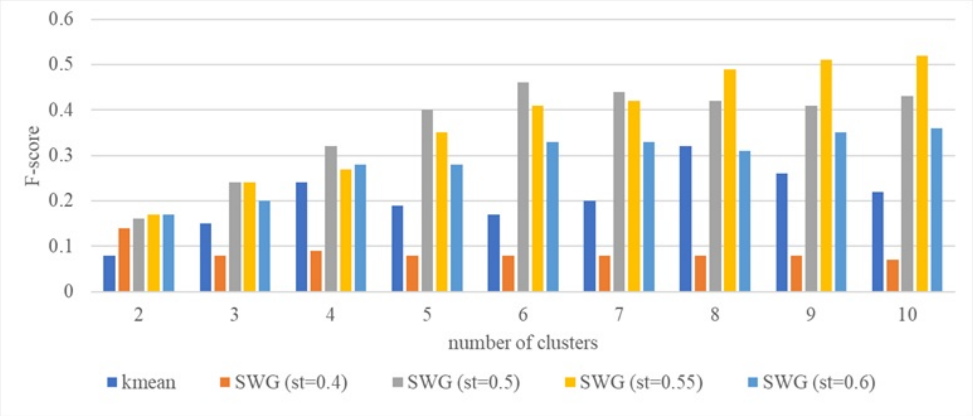

Fig. 10. Quality of clustering of our algorithm compared with k-means algorithm on searchsnippets dataset.

The STC algorithm produced more mixed clusters, capturing terms related to Health Drug, Sports Wins, Business Prices, and Technology Google.

Our proposed method, SWG, demonstrated a broader and more topic-specific range:

Cluster#1: Software and Development — “Microsoft”,

TABLE IV. Representative Words of Clusters Produced by each Competitive Algorithm on AG News Dataset

Cluster No.	Lingo	STC	SWG
1	Drug	Drug	Microsoft, Google, IBM, Linux, Windows, Intel
2	Wins	Wins	China, UK, Olympic, US, Sox, Phelps
3	U.S.	U.S.	Profit, Stocks, Prices, Oil, Costs, Trial
4	Wins Gold	Prices	Cancer, Study, Drug, Risk, Health, Disease
5	Cancer Drug	Google	Help, gets, takes, Slow, unveils, Higher
6	Phelps Wins	Oil	Wins, Gets, Takes, Says, Finds, Helps

TABLE V. Representative Words of Clusters Produced by each Competitive Algorithm on Tweet Dataset

Cluster No.	Lingo	STC	SWG
1	Boris Johnson Says	Year	people, company, secretary, workers, women, economic, children, business
2	PM Says	Boris Johnson	win, NBA, games, Lakers, team, playoffs
3	Labour Says	LIVE	Labour, MP, UK, Conservative, government, Boris
4	MP Says	Launch	spacecraft, NASA, Earth, astronauts, Mars, science

TABLE VI. Representative Words of Clusters Produced by each Competitive Algorithm on SearchSnippets Dataset

Cluster No.	Lingo	STC	SWG
1	News News	Edu	research, university, students, school, theory, physics
2	Research Research	News	science, disease, health, computer, biology
3	Wikipedia Wikipedia	Research	web, online, articles, photos, website, art
4	Edu Research	Wikipedia Wiki, Wikipedia Encyclopedia	world, political, gov, games, tournament, election
5	Information News	Gov	movie, film, music, books, band, guitar
6	News Sports	Articles	buy, companies, market, department, industry, organizations
7	Articles News	Computer	linux, wiki, microsoft, cpu, unix, ibm
8	Information Research	Resources	sports, football, soccer, tennis, basketball, president

"Google", "IBM", "Linux", "Windows", "Intel"

Cluster#2: Sports — "China", "UK", "Olympic", "US", "Sox", "Phelps"

Cluster#3: Business — "Profit", "Stocks", "Prices", "Oil", "Costs", "Trial"

Cluster#4: Health — "Cancer", "Study", "Drug", "Risk", "Health", "Disease"

Clusters#5 and Cluster#6: Contained more general and ambiguous terms, making topic identification less conclusive.

These results indicate that the SWG algorithm is more effective at generating semantically coherent cluster labels compared to the baseline algorithms, particularly in identifying multi-category topics in the AG News dataset.

For the results on the Tweet dataset, as presented in Table V, the dataset included four predefined topics: $Politics$, $Business$, $Science/Technology$, and $Sports$.

The Lingo algorithm focused almost exclusively on the Politics topic across all clusters, lacking clear differentiation between other topical categories.

The STC algorithm captured the Politics topic in Cluster#2, while the remaining clusters (Clusters#1, Cluster#3, and Cluster#4) contained more ambiguous or generic terms, such as "Year" "LIVE" and "Launch" which made topic interpretation less definitive.

In contrast, our proposed SWG algorithm successfully identified three distinct topics. Table VI shows respresentarive words of clusters produced by each competitive algorithm on searchSnippets dataset.

VI. Conclusion

In this study, we focused on short text clustering with two main objectives: providing high-quality results by grouping together short texts with similar content and generating meaningful cluster labels to aid user understanding.

From the results in section of Quality of Clustering, our proposed algorithm outperformed competitive algorithms across various key parameter variations. The Semantic Word Graph consistently produced significantly better clustering quality.

Furthermore, the representative words generated by the Semantic Word Graph algorithm demonstrated its ability to appropriately represent topics in a cluster. These words struck a balance between being overly specific and excessively general. The cluster labels derived from the Semantic Word Graph algorithm covered multiple topics within the initial clusters, as evidenced by the results in the Result and Discussion section.

The sensitivity analysis conducted on our proposed algorithm offered valuable insights into its performance under different parameter settings. Specifically, by varying the similarity threshold and observing the resulting clustering output, we gained a deeper understanding of the algorithm's response to input variations.

We observed that increasing the similarity threshold within the subgraph detection step improved the algorithm's performance until an optimal threshold was reached. However, setting the similarity threshold too high led to a slight decrease in clustering performance. Conversely, reducing the similarity threshold resulted in more general semantic connections and an increase in the number of documents per base cluster, thereby impacting the clustering performance. Consequently, determining an appropriate value for the similarity threshold is critical and should be tailored to the specific application and characteristics of the input documents.

In conclusion, our study demonstrated the effectiveness of the proposed Semantic Word Graph algorithm in achieving high-quality clustering results and generating meaningful cluster labels for short texts. The sensitivity analysis provided insights into parameter settings, further enhancing the algorithm's performance and customization capabilities.

Future research could focus on enhancing the SWG framework by integrating advanced contextual embeddings, such as BERT or Sentence-BERT, to capture richer and more nuanced semantic relationships between words. In addition, developing adaptive methods for automatically determining optimal similarity thresholds would allow the algorithm to adjust dynamically to the characteristics of different datasets, improving clustering quality without extensive manual tuning. Finally, incorporating user-centric evaluation—through usability studies and real-world testing—would provide insights into how well the generated cluster labels support human understanding and decision-making, ensuring that the method is not only effective in quantitative terms but also practical and interpretable for end users.

References

[1] P. Gurung and R. Wagh, "A study on topic identification using k means clustering algorithm: Big vs. small documents," *Advances in Computational Sciences and Technology*, vol. 10, no. 2, pp. 221–233, 2017.

[2] C. Clifton and R. Cooley, "Topcat: Data mining for topic identification in a text corpus," in *Proc. PKDD 1999: Principles of Data Mining and Knowledge Discovery*, 1999, pp. 174–183.

[3] Z. Ghaemi and M. Farnaghi, "Event detection from geotagged tweets considering spatial autocorrelation and heterogeneity," *Journal of Spatial Science*, vol. 66, no. 3, pp. 353–371, 2021.

[4] J. Weng and B. Lee, "Event detection in twitter," in *Proceedings of the AAAI 2011 International Conference on Web and Social Media*, 2011, pp. 401–408.

[5] S. Behpour, M. Mohammadi, M. V. Albert, Z. S. Alam, L. Wang, and T. Xiao, "Automatic trend detection: Time-biased document clustering," *Knowledge-Based Systems*, vol. 220, p. 106907, 2021.

[6] X. Chen, S. Duan, and L. Wang, "Research on clustering analysis of internet public opinion," *Cluster Computing*, vol. 22, pp. 5997–6007, 2019.

[7] S. B. Kaleel and A. Abhari, "Cluster-discovery of twitter messages for event detection and trending," *Journal of Computational Science*, vol. 6, no. 1, pp. 47–57, 2015.

[8] I. Beregovskaya and M. Koroteev, "Review of clustering-based recommender systems," arXiv preprint arXiv:1705.06157, 2017, available from: https://arxiv.org/abs/1705.06157.

[9] Y. Betancourt and S. Ilarri, "Use of text mining techniques for recommender systems," in *Proceedings of the International Conference on Enterprise Information Systems (ICEIS)*, 2020, pp. 780–787.

[10] M. Kozlowski and H. Rybinski, "Clustering of semantically enriched short texts," *Journal of Intelligent Information Systems*, vol. 53, pp. 69–92, 2018.

[11] J. Xu, B. Xu, P. Wang, P. Zheng, S. Tian, and J. Zhao, "Self-taught convolutional neural networks for short text clustering," *Neural Networks*, vol. 88, pp. 22–31, 2017.

[12] Y. Kim, "Convolutional neural networks for sentence classification," arXiv preprint arXiv:1408.5882, 2022, available from: https://arxiv.org/abs/1408.5882.

[13] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Proceedings of the Neural Information Processing Systems Conference (NeurIPS)*, 2013, pp. 3111–3119.

[14] Google Inc., "Google code archive," https://code.google.com/archive/p/word2vec/, 2025, [Online; cited 2025 Mar 12].

[15] V. D. Blondel, J. L. Guillaume, R. Lambiotte, and E. Lefebvre, "Fast unfolding of communities in large networks," *Journal of Statistical Mechanics: Theory and Experiment*, vol. 2008, no. 10, p. P10008, 2008.

[16] S. Osiński, J. Stefanowski, and D. Weiss, "Lingo: Search results clustering algorithm based on singular value decomposition," in *Intelligent Information Processing and Web Mining*, ser. Advances in Soft Computing, 2004, vol. 25, pp. 359–368.

[17] O. Zamir and O. Etzioni, "Web document clustering: A feasibility demonstration," in *Proceedings of the ACM SIGIR Conference on Research and Development in Information Retrieval*, 1998, pp. 46–54.

[18] O. Etzioni and O. Zamir, "Grouper: A dynamic clustering interface to web search results," in *Proceedings of the 8th International Conference on World Wide Web*, 1999, pp. 1361–1374.

[19] X. Phan, L. Nguyen, and S. Horiguchi, "Learning to classify short and sparse text and web with hidden topics from large-scale data collections," in *Proceedings of the 17th International Conference on World Wide Web*, 2008, pp. 91–100.

[20] Y. Tay, M. Dehghani, J. P. Gupta, V. Aribandi, D. Bahri, Z. Qin, and et al., "Are pre-trained convolutions better than pre-trained transformers?" in *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing*, 2021, pp. 4349–4359.

[21] U. Scaiella, P. Ferragina, A. Marino, and M. Ciaramita, "Topical clustering of search results," in *Proceedings of WSDM 2012: International Conference on Web Search and Data Mining*, 2012, pp. 223–232.

[22] M. Yuan, P. Lin, and J. Zobel, "Document clustering vs topic models: A case study," in *Proceedings of the 25th Australasian Document Computing Symposium*, 2021, pp. 1–8.

[23] K. K. Bharti and P. K. Singh, "Hybrid dimension reduction by integrating feature selection with feature extraction method for text clustering," *Expert Systems with Applications*, vol. 42, pp. 3105–3114, 2015.

[24] Y. Fan, L. Shi, and L. Yuan, "Topic modeling methods for short texts: A survey," *Journal of Intelligent & Fuzzy Systems*, vol. 45, pp. 1971–1990, 2023.

[25] D. Crabtree, X. Gao, and P. Andreae, "Improving web clustering by cluster selection," in *Proceedings of the IEEE/WIC/ACM International Conference on Web Intelligence*, 2005, pp. 172–178.

[26] S. Lazemi, H. Ebrahimpour-Komleh, and N. Noroozi, "Improving persian dependency-based semantic role labeling using semantic and structural relations," in *Proceedings of the 4th International Conference on Pattern Recognition and Image Analysis*, 2019, pp. 163–167.