Autoencoder based centrality for topic representative term extraction in short text clustering

Tapan Mahata[0009-0007-7843-115X],
Abhradeep Datta[0000-0003-3514-5771],
Debanga Raj Neog[0000-0002-2794-4787], and
Ashok Singh Sairam[0000-0001-9527-6496]

Indian Institute of Technology Guwahati, Assam, India
{t.mahata,a.datta,dneog, ashok }@iitg.ac.in

Abstract. Short text clustering is crucial for extracting knowledge from online social media platforms. The task is challenging due to the limited text length and various forms of noise. Short texts about a specific topic often share common terms that are representative of a topic, known as topic-representative keywords. The goal is to identify these keywords by clustering at the word level. The short texts are represented as a word graph network where nodes represent words and edges represent co-occurrence or semantic relationships, capturing the context. We propose a novel centrality measure, Encoder Indexed Centrality (EIC), that is applied to the graph to identify the topic-representative keywords. The proposed metric is based on the latent representation of existing edge-weighted centrality measures. By measuring thematic closeness and semantic coherence, we discover semantically relevant word clusters around each keyword. This induced word cloud serves as the basis for short text clustering. Our model, implemented on a short-text Twitter dataset, demonstrates superior performance compared to various baselines.

Keywords: Short Text Clustering · Keyword Extraction · Graph Centrality.

1 Introduction

Social media platforms have revolutionized communication among people in the form of shared interests and opinions about various social events. Text clustering, an unsupervised learning technique is primarily used to cluster the social media text and extract the topic-representative keywords. However, short text clustering comes with its own set of challenges. Short texts create sparse feature vectors with TF-IDF (Term Frequency-Inverse Document Frequency) [13] or bag-of-words [8], complicating pattern recognition for clustering algorithms. Despite fewer words, a large vocabulary leads to high-dimensional feature spaces and the curse of dimensionality. Additionally, short texts are often ambiguous and noisy, lacking the context of longer documents, making it difficult to understand the meaning and to perform clustering.

Short-text clustering methods can be broadly categorized into three categories: distance-based, topic-modeling, and deep learning-based. Distance-based clustering uses vector representations like bag-of-words and TF-IDF, applying distance metrics (e.g., Euclidean, Cosine, Manhattan, Pearson, etc.) to group texts. Topic modeling clusters texts by identifying latent topics using techniques like LDA (Latent Dirichlet Allocation) [2], LSA (Latent Semantic Analysis) [3]. NMF (Non-negative Matrix Factorization) [17] is a popular technique used in topic modeling and document clustering. Deep learning approaches involve word embeddings from language models and semantic composition to represent phrases or sentences, enabling more effective short-text grouping.

However, the above-mentioned clustering paradigms come with their own set of challenges. The high dimensional feature space of the vector space model [14] fails to capture the rich sequential and contextual information of the data, especially short texts. Topic modeling lacks clear interpretability, complex parameter adjustment, and poor representation of short text. Topic models depend upon the rich co-occurrence of words in documents which is absent in short texts. Deep learning-based contextual embedding like BERT (Bidirectional Encoder Representations from Transformers) [4] solves the problem to some extent but comes with a high computational cost and is resource-intensive. The performance depends on the quality and diversity of the training data.

In this research, we overcome the challenge of short text clustering by low resource graph-based method to discover topic representative keywords. The motivation is that short texts, being syntactically incomplete and less contextual, often share certain common terms that can effectively represent a topic. These inherent topics can be mapped to a cluster of short texts. We use the concept of graph centrality to mark the topic-representative keywords and also formulate a mechanism to discover semantically relevant word clusters around every topic-representative keyword. Our main contributions are as follows:

- We aim to solve the problem of short text clustering at the word level by developing a word graph network where nodes represent unique words in the corpus and edges represent co-occurrence or semantic relationships between words, capturing the context.
- We use an auto-encoder-based [7] dimension reduction technique to create a unique centrality measure called EIC (Encoder Indexed Centrality), which is based on the latent representation of existing edge-weighted centrality measures (Betweenness, Closeness, Degree, Eigenvector, Clustering Coefficient, PageRank).
- EIC (Encoder Indexed Centrality) is applied on the word graph network with a window size of three to determine the topic representative keywords.
- We discover closely bound semantically relevant word clusters around every topic-representative keyword by measuring thematic closeness and semantic coherence. This induced word cloud around each topic representative keyword acts as a base for short text clustering.

2 Related Works

The existing short-text clustering research can be reviewed in three categories: distance-based clustering, topic modeling, and deep learning-based clustering.

2.1 Distance Based Clustering

This method groups text vectors using distance metrics and clustering algorithms. The vector space model [14] represents documents as high-dimensional vectors, with TF-IDF highlighting term importance. To overcome the sparseness of using a vector space model in short texts, word embedding methods enhance feature representation. Clustering algorithms like k-means, hierarchical clustering, and DBSCAN are applied to these features. The choice of distance measure, such as Euclidean, Cosine, Manhattan, or Pearson, significantly impacts clustering performance, cluster shape, and result interpretation.

The high-dimensional feature space of the vector space model often fails to capture the sequential and contextual richness of short texts. This issue is mitigated by enhancing semantic representation using external knowledge bases like Wikipedia[1] [1] [12] [16], which address the sparsity of raw word features. For instance, Banerjee *et al.* [1] augmented TF-IDF representations with relevant Wikipedia concepts, while Zheng *et al.* [19] enriched short text features by mapping documents to a hidden semantic space, adding virtual term frequencies of new words to improve clustering performance.

2.2 Topic Modeling

Topic modeling-based clustering is a method that groups short texts by identifying underlying topics within a collection of documents. Unlike traditional clustering which directly groups documents based on similarity measures, topic modeling first identifies latent topics and then clusters documents based on these topics. LDA [2], LSA [3], and NMF [17] are some of the popular topic modeling techniques. However, they lack clear interpretability, complex parameter adjustment, and poor representation of short text. Also, the success of topic models depends upon the rich co-occurrence of words in documents which is absent in short texts.

Yin *et al.* [18] presented GSDMM for short text clustering, using the Dirichlet Multinomial Mixture (DMM) model. GSDMM is a probabilistic generative model for the short text corpus. It addresses the sparsity issue of short texts in clustering tasks by assuming that each short text is created by a single latent topic. The themes of brief texts are determined using Collapsed Gibbs sampling methods and utilized as cluster labels.

[1] https://www.wikipedia.org/

2.3 Deep Learning-based Clustering

Deep learning has advanced short text grouping through two main approaches: word embeddings and semantic composition at the sentence level. Word embeddings like Word2Vec [9], GloVe [11], and FastText [6] train language models to represent words as dense, low-dimensional vectors, forming semantic clusters. However, they treat words in isolation, neglecting context and sentence structure. Semantic composition models like BERT address these limitations by generating contextual embeddings, but they are computationally intensive and dependent on the quality of training data.

To tackle short text clustering, we propose a low-resource method combining graph-based techniques with deep learning. Instead of clustering documents, we focus on identifying topic-representative keywords. We model this by creating a word graph network, where nodes are words and edges represent word co-occurrences. A novel centrality measure, Encoder Indexed Centrality (EIC), is introduced, integrating various edge-weighted centrality measures (Betweenness, Closeness, Degree, Eigenvector, Clustering Coefficient, PageRank) using an auto-encoder-based dimension reduction technique. EIC combines multiple centralities into a single index, providing a superior descriptor for keyword extraction.

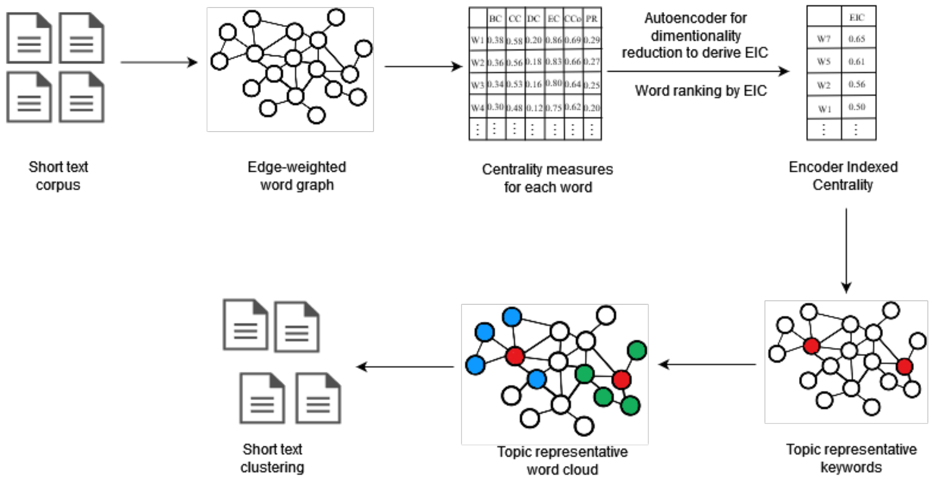
3 Methodology

We first introduce the terminologies and notations in our proposed algorithms. A corpus D is a collection of m documents denoted as $D = \{d_1, d_2, d_3, \ldots, d_m\}$. A document, denoted by d, is a sequence of n terms, $d = \{w_1, w_2, \ldots, w_n\}$, where w_i is the ith term in d. Given a short text corpus D with m documents, the problem of short text clustering is to partition D into N different groups, in which short texts in the same group are more similar to each other than those in other groups.

The computational framework of our proposed short text clustering approach using topic-representative keywords is depicted in Fig.1. First, we construct an edge-weighted word graph and compute various centrality measures for the nodes. These measures are integrated into a single metric, Enhanced Integrated Centrality (EIC), to identify the top words as topic-representative keywords. Clusters are then formed around these keywords by finding semantically relevant words. The resulting word clouds serve as the basis for clustering the short texts.

3.1 Edge Weighted Word Graph

The documents are tokenized into individual words, forming a vocabulary used to construct the word graph network $G = (V, E)$, where V consists of unique words and E represents corpus-level co-occurrence frequencies between two corresponding words within a window size of 3. A smaller window size is used to maintain precision, as words further apart contribute less to the document's semantics [15].


Short text corpus
Edge-weighted word graph
Centrality measures for each word
Autoencoder for dimentionality reduction to derive EIC
Word ranking by EIC
Encoder Indexed Centrality
Topic representative keywords
Topic representative word cloud
Short text clustering

	BC	CC	DC	EC	CCo	PR
W1	0.38	0.58	0.20	0.86	0.69	0.29
W2	0.36	0.56	0.18	0.83	0.66	0.27
W3	0.34	0.53	0.16	0.80	0.64	0.25
W4	0.30	0.48	0.12	0.75	0.62	0.20
⋮	⋮	⋮	⋮	⋮	⋮	⋮

	EIC
W7	0.65
W5	0.61
W2	0.56
W1	0.50
⋮	⋮

Fig. 1: Short text clustering architecture using topic representative keywords.

3.2 Node Ranking by Centrality Measures

We aim to identify the most important nodes in an edge-weighted word graph. Centrality measures rank the importance of vertices based on their capacity for influence. We use six centrality measures to rank the nodes: Degree, Betweenness, Closeness, Eigenvector, Clustering coefficient, and PageRank. Each measure focuses on different network parameters, such as local information flow, network load, and regional connections. Below, we briefly discuss each of these measures.

- Degree centrality: The weighted degree centrality $C_D(v)$ is given by:

  $$C_D(v) = \sum_{u \in N(v)} w_{vu}$$

  where w_{vu} is the edge weight between nodes v and u, and $N(v)$ is the set of neighbors of v.
- Betweenness centrality: The weighted betweenness centrality $C_B(v)$ is given by:

  $$C_B(v) = \sum_{s \neq v \neq t} \frac{\sigma_{st}(v)}{\sigma_{st}}$$

  where σ_{st} is the total number of shortest paths from node s to node t, and $\sigma_{st}(v)$ is the number of those paths that pass through v.
- Closeness centrality: The weighted closeness centrality $C_C(v)$ is given by:

  $$C_C(v) = \frac{1}{\sum_{u \in V} d(v,u)}$$

  where $d(v,u)$ is the shortest path distance between nodes v and u, taking into account the edge weights.

- Eigenvector centrality: The eigenvector centrality x_v is given by:

$$x_v = \frac{1}{\lambda} \sum_{u \in N(v)} w_{vu} x_u$$

where x_v is the eigenvector centrality of the node v, w_{vu} is the weight of the edge between nodes v and u, and λ is a constant (the largest eigenvalue of the adjacency matrix).
- Clustering Coefficient: The weighted clustering coefficient $C(v)$ is given by:

$$C(v) = \frac{1}{d(v)(d(v)-1)} \sum_{u,w \in N(v)} \frac{w_{vu} + w_{vw}}{2}$$

where $d(v)$ is the degree of node v and w_{vu}, w_{vw} are the weights of the edges.
- PageRank: The PageRank $PR(v)$ is given by:

$$PR(v) = \frac{1-d}{N} + d \sum_{u \in M(v)} \frac{PR(u) \cdot w_{uv}}{\sum_{w \in N(u)} w_{uw}}$$

where d is the damping factor, N is the total number of nodes, $M(v)$ is the set of nodes that link to v, $N(u)$ is the set of nodes that u links to, and w_{uv} is the weight of the edge from node u to node v.

3.3 Encoder Indexed Centrality

Centrality measures rank network nodes from different perspectives, often leading to conflicts. For instance, a node with a high degree of centrality might have low betweenness centrality if it isn't a key connector between different groups, and vice versa. No single measure is universally best for keyword extraction. To address this, we propose Encoder Indexed Centrality (EIC), which combines latent representations of multiple edge-weighted centrality measures into a single index, providing a comprehensive view of node importance.

Fig. 2 illustrates the autoencoder architecture used in the EIC model. The encoder has an input layer dimension of 6, with hidden layers of 36, 216, 512, 216, and 36 neurons, all using ReLU activation. The latent space is 1-dimensional with a linear activation layer. The decoder mirrors the encoder, with an input layer dimension of 1, hidden layers of 36, 216, 512, 216, and 36 neurons using ReLU activation, and an output layer dimension of 6 with a sigmoid activation function.

The 6-dimensional input data is processed through the encoder's hidden layers, which learn complex relationships among the centrality features. The encoder compresses the data to a 1-dimensional latent space, retaining only the essential features. The decoder then reconstructs the original 6-dimensional data from this latent feature. Using ReLU activation functions, the autoencoder captures non-linear mappings and complex patterns in the data. After training, the encoder is used to transform the 6-dimensional input into the 1-dimensional latent feature, called Encoder Indexed Centrality (EIC). The top words ranked by EIC are termed as Encoder Indexed Keywords (EIK).

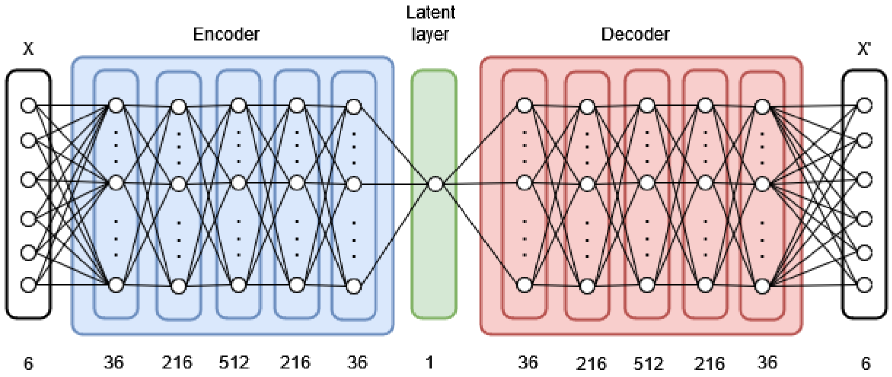


Fig. 2: Encoder Indexed Centrality.

3.4 Topic Representative Word Cloud

After obtaining the Encoder Indexed Keywords (EIK), we discover closely bound semantically relevant words. Algorithm 1 depicts the formation of topic representative word cloud. The input to the algorithm is the word graph (G), the Encoder Indexed Keywords (EIK), and k, the number of predetermined topics to identify. The algorithm iterates through EIK. For each keyword in EIK, if it has not been visited before, the $adjacent(word)$ function is used to discover its adjacent keywords. These adjacent keywords along with the current keyword are then added to a word cloud (t), which represents the word cloud for the current keyword. Both the current keyword and its adjacent keywords are marked as visited to avoid redundancy. The process continues until the word cloud set (T) contains at most k topics.

EIK act as potential candidate keys representing the centroid of the word cloud and their corresponding one-hop neighbors with decreasing order of edge weight are the supporting keys of the word cloud. These terms are thematically close and semantically relevant to the topic representative keywords (centroid).

3.5 Short text clustering

The induced word cloud around each topic representative keyword (centroid) acts as a base for short text clustering. Each word cloud can be interpreted as a topic.

Let, n be the total number of topic representative word clouds, $T = \{t_1, t_2, \ldots, t_n\}$, where t_i is a topic or word cloud and T is word cloud set. We define the mapping between a short text and a word cloud by considering the length of their overlapped terms.

We assign each of the m short text $d_i \in D$, $1 \leq i \leq m$, to a word cloud t_j if d_i has the maximum number of common unigrams with the topic representative word cloud t_j than any other word clouds. The following equation depicts the

Algorithm 1 Topic Representative Word Cloud Construction

```
**Input:** $G = (V, E)$, $EIK$: Encoder Indexed Keywords, $k$: number of topics
**Output:** $T$: word cloud set
**Init:** $T \leftarrow$ NULL, $visited \leftarrow$ NULL
**for each** $keyword \in EIK$ **do**
    **if** $word \notin visited$ **then**
        $t \leftarrow adjacent(keyword) \cup \{keyword\}$
        $T \leftarrow T.append(t)$
        $visited \leftarrow visited.append(t)$
        **if** length($T$) $\geq k$ **then**
            **break**
        **end if**
    **end if**
**end for**
**return** $T$
```

topic assignment for short texts:

$$\arg\max_{t_j \in T} |d_i \cap t_j|, \tag{1}$$

where $|d_i|$ indicates the number of words in document d_i, $|d_i \cap t_j|$ denotes the number of common words shared between d_i and t_j.

4 Experiment Details

In this section, we present a detailed analysis of the dataset used, the baseline methods to be compared, and evaluation metrics.

4.1 Dataset

The Twitter dataset used in our experiment is taken from [10] with 7.67 million tweets related to 6 different crisis events. However, 32,462 tweets are labeled with ground truth topics which include Sandy Hurricane, Alberta Floods, Boston Bombings, Oklahoma Tornado, Queensland Floods, and West Texas Explosion. The number of labeled tweets across the mentioned topics is given in Table 1.

4.2 Baseline Models

We compare our proposed approach with the below-mentioned methods for short text clustering:

- LDA [2] is a generative probabilistic model used for topic modeling. It models documents as mixtures of topics, while topics are probability distributions over words. Each document has a topic distribution vector derived from word occurrence patterns. The k-means algorithm is used on the topic distribution for clustering tasks.

Table 1: Dataset Description.

Topic	**Number of labelled tweets**
Sandy Hurricane	6138
Alberta Floods	5189
Boston Bombings	5648
Oklahoma Tornado	4827
Queensland Floods	5414
West Texas Explosion	5246

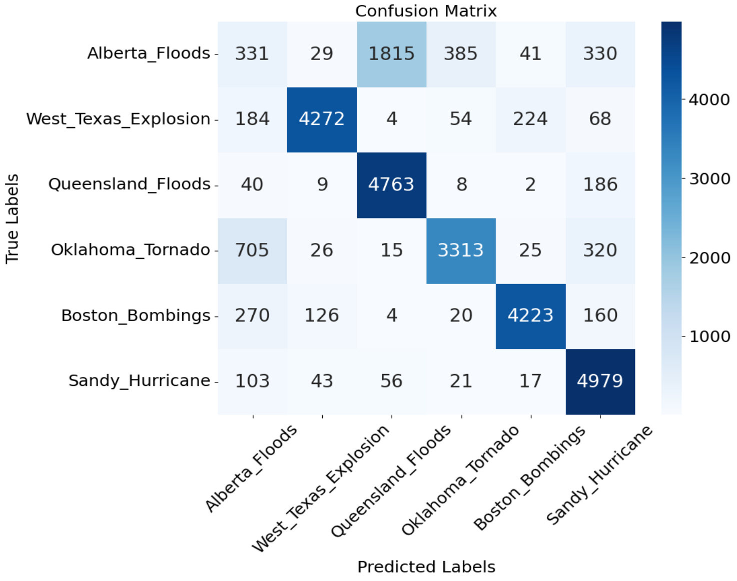


Fig. 3: Confusion Matrix for EIC.

- NMF [17] in topic modeling is used to discover latent topics within a document collection. It is a dimension reduction technique that decomposes a document-term matrix into two lower-dimensional matrices representing the relationships between documents and topics, and topics and terms, respectively.
- GSDMM (Gibbs Sampling Dirichlet Multinomial Mixture) [18] is a model designed for clustering short texts. It is especially beneficial for short text data that frequently lacks word co-occurrence information. The method works on the assumption that each short text is generated from a single latent topic and uses Gibbs sampling to get the topic index. Short texts related to the same latent topic are organized into clusters.

4.3 Evaluation Metrics

In our research, we use a short text dataset labeled with ground truth topics. For evaluation, we automatically curate the set of ground truth keywords to better describe specific topics. We then evaluate how well the detected keywords from our topic model match these ground truth keywords. The automated addition of ground truth keywords is done using the KeyBERT model [5], parameterized with the respective Wikipedia article for each topic.

We evaluate our algorithm by comparing the ground truth topics with the detected topics. Ground truth topics are augmented with keywords from Wikipedia. Topic mapping is performed by matching keywords between each pair of ground truth and detected topics. After mapping, tweets are assigned to detected topics using Equation (1). We use Precision, Recall, and F1-score to assess clustering accuracy, evaluating the relevance and accuracy of grouping tweets based on the discovered topics compared to the ground truth.

Table 2: Ground truth topic with keywords from Wikipedia.

Topic	**Representative Keyword Group**
Sandy Hurricane	‘hurricane’, ‘cyclone’, ‘sandy’, ‘landfall’, ‘storm’, ‘bahamas’, ‘caribbean’, ‘superstorm’, ‘cuba’, ‘tropical’, ‘jamaica’, ‘coast’, ‘coastal’, ‘atlantic’, ‘haiti’
Alberta Floods	‘floods’, ‘flooding’, ‘alberta’, ‘flooded’, ‘disaster’, ‘evacuation’, ‘saskatchewan’, ‘wildfire’, ‘fort’, ‘rainfall’, ‘provincial’, ‘emergency’, ‘catastrophic’, ‘damages’, ‘canadian’
Boston Bombings	‘tsarnaev’, ‘shootout’, ‘brothers’, ‘killed’, ‘suspects’, ‘terrorist’, ‘boston’, ‘marathon’, ‘bombs’, ‘bomb’, ‘bombing’, ‘shot’, ‘explosive’, ‘watertown’, ‘tamerlan’
Oklahoma Tornado	‘tornado’, ‘tornadoes’, ‘storms’, ‘oklahoma’, ‘thunderstorm’, ‘windy’, ‘winds’, ‘moore’, ‘thunderstorms’, ‘storm’, ‘hurricane’, ‘meteorological’, ‘wind’, ‘ef5’, ‘weather’
Queensland Floods	‘cyclone’, ‘bundaberg’, ‘evacuated’, ‘storms’, ‘flooding’, ‘queensland’, ‘rainfall’, ‘mundubbera’, ‘tropical’, ‘rescues’, ‘damage’, ‘affected’, ‘bay’, ‘impacted’, ‘burnett’
West Texas Explosion	‘ammonium’, ‘explosion’, ‘exploded’, ‘fertilizer’, ‘explosives’, ‘nitrate’, ‘facility’, ‘waco’, ‘texas’, ‘west’, ‘emergency’, ‘killed’, ‘destroyed’, ‘damaged’, ‘firearms’

4.4 Result Analysis

This section explains the analysis of short-text clustering and the quality of topic-term discovery of short-text.

Analysis of short text clustering From Table 3, we observe that our proposed model achieves the best clustering in terms of almost all metrics for short text.

Our model beats GSDMM, which is the highest-ranked clustering algorithm, by 6.8% in recall and 6.20% in F1-score. However, GSDMM exhibits a slightly higher precision (1.2%) score due to the fact that GSDMM explicitly models the distribution of topics within clusters and the distribution of words within topics. This dual-level modeling aids in the formation of more exact clusters by taking into account both document-topic and word-topic correlations. Our model beats NMF by a margin of 13% in precision, 6.23% in recall, 11.6% in F1-score. We observe that LDA is the least effective in short text clustering as it requires long text where topics can be distributed over many words, and also decent word co-occurrence. The confusion matrix for calculation of the aforementioned metrics for EIC is represented in Fig 3.

Table 3: Comparison of the proposed model with other clustering models.

Model	**Precision**	**Recall**	**F1-score**
LDA [2]	0.4869	0.4715	0.4742
NMF [17]	0.6620	0.7120	0.6685
GSDMM [18]	**0.7579**	0.7084	0.7024
Our Model	0.7489	**0.7564**	**0.7459**

Table 4: Topic Representative Word Cloud from LDA.

Topic	**Representative Keyword Group**
Sandy Hurricane	‘hurricane’, ‘sandy’, ‘im’, ‘like’, ‘dont’, ‘help’, ‘get’
Alberta Floods	‘flood’, ‘tornado’, ‘storm’, ‘australia’, ‘amazing’, ‘woman’, ‘oklahoma’
Boston Bombings	‘boston’, ‘marathon’, ‘bombing’, ‘suspect’, ‘bomb’, ‘fbi’, ‘explosion’
Oklahoma Tornado	‘oklahoma’, ‘prayer’, ‘thought’, ‘affected’, ‘go’, ‘everyone’, ‘boston’
Queensland Floods	‘flood’, ‘queensland’, ‘crisis’, ‘australia’, ‘death’, ‘alberta’, ‘toll’
West Texas Explosion	‘explosion’, ‘texas’, ‘plant’, ‘fertilizer’, ‘west’, ‘waco’, ‘video’

Analysis of topic term discovery In this section, we analyze the capacity to discover effective topic-representative keywords by LDA, NMF, and GSDMM as compared to our model. For each model, we represent the topics (Sandy Hurricane, Alberta Floods, Boston Bombings, Oklahoma Tornado, Queensland Floods, and West Texas Explosion) with the top-ranked generated keywords. Table 2 depicts the ground truth keywords for each topic. Tables 4-6 show the resultant topic representative keywords by LDA, NMF, and GSDMM.

Table 5: Topic Representative Word Cloud from NMF.

Topic	**Representative Keyword Group**
Sandy Hurricane	‘hurricane’, ‘sandy’, ‘im’, ‘like’, ‘name’, ‘safe’, ‘everyone’
Alberta Floods	‘baby’, ‘led’, ‘stuffed’, ‘hoisted’, ‘powerful’, ‘bag’, ‘storm’
Boston Bombings	‘boston’, ‘marathon’, ‘bombing’, ‘suspect’, ‘fbi’, ‘bomb’, ‘victim’
Oklahoma Tornado	‘oklahoma’, ‘prayer’, ‘tornado’, ‘thought’, ‘go’, ‘affected’, ‘everyone’
Queensland Floods	‘flood’, ‘queensland’, ‘australia’, ‘rise’, ‘water’, ‘crisis’, ‘photo’
West Texas Explosion	‘explosion’, ‘texas’, ‘plant’, ‘fertilizer’, ‘west’, ‘waco’, ‘video’

Table 6: Topic Representative Word Cloud from GSDMM.

Topic	**Representative Keyword Group**
Sandy Hurricane	‘hurricane’, ‘sandy’, ‘im’, ‘like’, ‘people’, ‘boston’, ‘get’
Alberta Floods	‘flood’, ‘queensland’, ‘calgary’, ‘australia’, ‘alberta’, ‘water’, ‘flooding’
Boston Bombings	‘boston’, ‘bombing’, ‘marathon’, ‘suspect’, ‘flood’, ‘queensland’, ‘bomb’
Oklahoma Tornado	‘oklahoma’, ‘tornado’, ‘prayer’, ‘help’, ‘victim’, ‘boston’, ‘affected’
Queensland Floods	‘flood’, ‘australia’, ‘queensland’, ‘crisis’, ‘rise’, ‘water’, ‘toll’
West Texas Explosion	‘explosion’, ‘texas’, ‘plant’, ‘fertilizer’, ‘west’, ‘waco’, ‘boston’

We observe that EIC model (Table 7) can better uncover topic-relevant keywords as compared to the other models (Tables 4-6).

- Sandy Hurricane: The resultant keywords for the topic “Sandy Hurricane” are almost comparable across all models. The inherent nature of these models can identify the most salient terms in the corpus that distinguish the given topic from the rest.
- Alberta Floods: The topics “Alberta Floods” and “Queensland Floods” consist of tweets about floods and hence, have overlapping keywords. Except for EIC, all the other models capture certain incorrect keywords. For example, LDA wrongly represents “Alberta Floods” with “australia” and “oklahoma” wherein “australia” should be attached with “Queensland Floods” and “oklahoma” with “Oklahoma Tornado”. Similarly, GSDMM wrongly interprets “Alberta Floods” by “queensland” and “australia” which should be mapped with “Queensland Floods”. NMF has a very weak representation of the same topic. Compared to the other models, EIC prevents the misrepresentation of the topics in case of “Alberta Floods”.
- Boston Bombings: GSDMM wrongly represents the topic ”Boston Bombings” with “flood” and “queensland” keywords. EIC and other models have a comparable keyword representation for the aforementioned topic.
- Oklahoma Tornado: For the topic “Oklahoma Tornado”, LDA and GSDMM misinterprets the topic with “boston” keyword which is related to “Boston

Table 7: Topic Representative Word Cloud from EIC model.

Topic	**Representative Keyword Group**
Sandy Hurricane	‘hurricane’, ‘sandy’, ‘like’, ‘affected’, ‘everyone’, ‘im’, ‘name’
Alberta Floods	‘people’, ‘rainfall’, ‘killed’, ‘prayer’, ‘dead’, ‘affected’, ‘go’
Boston Bombings	‘boston’, ‘marathon’, ‘bombing’, ‘suspect’, ‘explosion’, ‘bomb’, ‘victim’
Oklahoma Tornado	‘tornado’, ‘oklahoma’, ‘victim’, ‘affected’, ‘help’, ‘relief’, ‘moore’
Queensland Floods	‘flood’, ‘queensland’, ‘crisis’, ‘australia’, ‘water’, ‘victim’, ‘alberta’
West Texas Explosion	‘fertilizer’, ‘plant’, ‘explosion’, ‘texas’, ‘camera’, ‘west’, ‘caught’

Bombings”. EIC presents the most meaningful set of keywords in the form of “tornado”, “oklahoma”, “victim”, and “relief” for the concerned topic.

- Queensland Floods: EIC manages to prevent misrepresentation of topics in most cases except for the topic “Queensland Floods”.

We conclude that in most cases EIC successfully represents the topics with meaningful keywords as compared to the other models. The quality of EIC-based keywords is comparable with the ground truth keyword group for most of the topics. EIC-indexed keywords represent the thematic centroid of the word graph and the corresponding one-hop neighboring words with decreasing order of edge weight further add up to topic representative word cloud, enhancing semantic relevance among them.

5 Conclusion

Our research improves short-text clustering by identifying thematically enriched keyword groups that represent specific topics. Using an edge-weighted word graph network, words are ranked by Encoder Indexed Centrality (EIC) to identify key topics. Clusters form around these keywords based on thematic closeness and semantic coherence, creating a word cloud for clustering. Experiments on Twitter data show our method outperforms baseline models in clustering accuracy and topic term discovery.

In the future, we will try to solve the problem of topic misinterpretation when the same keywords fall under multiple topics with the help of word embedding models. We would experiment with multiple short text data of different topical domains to ensure the robustness of our model.

References

1. Banerjee, S., Ramanathan, K., Gupta, A.: Clustering short texts using wikipedia. In: Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval. pp. 787–788 (2007)

2. Blei, D.M., Ng, A.Y., Jordan, M.I.: Latent dirichlet allocation. Journal of machine Learning research **3**(Jan), 993–1022 (2003)
3. Deerwester, S., Dumais, S.T., Furnas, G.W., Landauer, T.K., Harshman, R.: Indexing by latent semantic analysis. Journal of the American society for information science **41**(6), 391–407 (1990)
4. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805 (2018)
5. Grootendorst, M.: Keybert: Minimal keyword extraction with bert. (2020)
6. Joulin, A., Grave, E., Bojanowski, P., Douze, M., Jégou, H., Mikolov, T.: Fasttext. zip: Compressing text classification models. arXiv preprint arXiv:1612.03651 (2016)
7. Liou, C.Y., Cheng, W.C., Liou, J.W., Liou, D.R.: Autoencoder for words. Neurocomputing **139**, 84–96 (2014)
8. Manning, C., Schutze, H.: Foundations of statistical natural language processing. MIT press (1999)
9. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781 (2013)
10. Olteanu, A., Castillo, C., Diaz, F., Vieweg, S.: Crisislex: A lexicon for collecting and filtering microblogged communications in crises. In: Proceedings of the international AAAI conference on web and social media. vol. 8, pp. 376–385 (2014)
11. Pennington, J., Socher, R., Manning, C.D.: Glove: Global vectors for word representation. In: Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP). pp. 1532–1543 (2014)
12. Phan, X.H., Nguyen, L.M., Horiguchi, S.: Learning to classify short and sparse text & web with hidden topics from large-scale data collections. In: Proceedings of the 17th international conference on World Wide Web. pp. 91–100 (2008)
13. Robertson, S.: Understanding inverse document frequency: on theoretical arguments for idf. Journal of documentation **60**(5), 503–520 (2004)
14. Salton, G., Wong, A., Yang, C.S.: A vector space model for automatic indexing. Communications of the ACM **18**(11), 613–620 (1975)
15. Vega-Oliveros, D.A., Gomes, P.S., Milios, E.E., Berton, L.: A multi-centrality index for graph-based keyword extraction. Information Processing & Management **56**(6), 102063 (2019)
16. Vo, D.T., Ock, C.Y.: Learning to classify short text from scientific documents using topic models with various types of knowledge. Expert Systems with Applications **42**(3), 1684–1698 (2015)
17. Xu, W., Liu, X., Gong, Y.: Document clustering based on non-negative matrix factorization. In: Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval. pp. 267–273 (2003)
18. Yin, J., Wang, J.: A dirichlet multinomial mixture model-based approach for short text clustering. In: Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining. pp. 233–242 (2014)
19. Zheng, C.T., Liu, C., San Wong, H.: Corpus-based topic diffusion for short text clustering. Neurocomputing **275**, 2444–2458 (2018)