Hindawi
Complexity
Volume 2020, Article ID 4750871, 11 pages
https://doi.org/10.1155/2020/4750871

Research Article

Sentence Embedding Based Semantic Clustering Approach for Discussion Thread Summarization

Atif Khan,[1] **Qaiser Shah**,[1] **M. Irfan Uddin**,[2] **Fasee Ullah**,[3] **Abdullah Alharbi**,[4] **Hashem Alyami**,[5] **and Muhammad Adnan Gul**[1]

[1]*Department of Computer Science, Islamia College Peshawar, Peshawar, KP, Pakistan*
[2]*Institute of Computing, Kohat University of Science and Technology, Kohat, Pakistan*
[3]*Department of Computer and Information Science, Faculty of Science and Technology, University of Macau, Macau, China*
[4]*Department of Information Technology, College of Computers and Information Technology, Taif University, Taif 21944, Saudi Arabia*
[5]*Department of Computer Science, College of Computers and Information Technology, Taif University, Taif 21944, Saudi Arabia*

Correspondence should be addressed to Atif Khan; atifkhan@icp.edu.pk and Fasee Ullah; faseekhan@gmail.com

Received 1 July 2020; Accepted 30 July 2020; Published 25 August 2020

Guest Editor: Furqan Aziz

Huge data on the web come from discussion forums, which contain millions of threads. Discussion threads are a valuable source of knowledge for Internet users, as they have information about numerous topics. The discussion thread related to single topic comprises a huge number of reply posts, which makes it hard for the forum users to scan all the replies and determine the most relevant replies in the thread. At the same time, it is also hard for the forum users to manually summarize the bulk of reply posts in order to get the gist of discussion thread. Thus, automatically extracting the most relevant replies from discussion thread and combining them to form a summary are a challenging task. With this motivation behind, this study has proposed a sentence embedding based clustering approach for discussion thread summarization. The proposed approach works in the following fashion: At first, word2vec model is employed to represent reply sentences in the discussion thread through sentence embeddings/sentence vectors. Next, K-medoid clustering algorithm is applied to group semantically similar reply sentences in order to reduce the overlapping reply sentences. Finally, different quality text features are utilized to rank the reply sentences in different clusters, and then the high-ranked reply sentences are picked out from all clusters to form the thread summary. Two standard forum datasets are used to assess the effectiveness of the suggested approach. Empirical results confirm that the proposed sentence based clustering approach performed superior in comparison to other summarization methods in the context of mean precision, recall, and *F*-measure.

1. Introduction

The content shared by Internet users in online forum platforms is a valuable repository of information. Since information and communication technologies (ICT) are rising at a high pace, a bulk of data is available online. Many users use web services to share their knowledge about specific subjects, which exist on the web in the form of discussion forum, blogs, or any other user generated content [1].

Discussion forums are also known as web forum, message boards, and bulletin boards. In the current era, discussion forums are becoming very popular because these platforms give easy access to users to share their information and allow them to discuss issues/topics of common interest. Huge data on the web come from discussion forums, which contain millions of threads. These threads are a valuable source of knowledge for Internet users as they have information about various topics. The threads, also called discussion threads, are important for those who post as well as for "lurkers," users who only read the replies. The discussion threads pertaining to single topic comprise a huge number of individual posts, which makes it hard for forum users to determine the most significant information in the thread.

A discussion thread is initiated when a user posts an initial post/question, and other users reply to that post, leading to an active discussion. As more and more users are involved in discussion, the number of replies for a given post within a thread increases, and this makes it difficult for users to read all the replies within a discussion thread [2].

In such a situation, the forum user will favor a short summary of the running discussion in order to get the idea of the long discussion thread in short time. It is hard for forum users to manually summarize the bulk of replies in the discussion thread. This necessitates an automatic solution for discussion thread summarization. With this motivation behind, this study has proposed a sentence embedding based clustering approach for discussion thread summarization. The approach will retrieve the most significant replies from the discussion thread to constitute a summary. Summarizing forum threads will assist the forum users to swiftly comprehend the main topic/idea of the discussion thread.

A few research studies applied both extractive and abstractive summarization techniques for discussion thread summarization such as e-mail threads summarization [3], summarizing written and spoken chats/conversations and meeting recordings [4], and summarizing Twitter topic [5].

Our approach for discussion thread summarization is different from prior approaches in few aspects. The approach is fully automatic and domain-independent, does not rely on any dictionary/resource, and does not involve any intervention of human to generate summary. It is a generic approach and is applicable to English discussion forum of any domain.

The suggested approach works in the following way: At first, we employed word2vec model to transform the collection of reply sentences into sentence embeddings/sentence vectors, which are obtained by averaging word embedding for each word in the reply sentences. Next, we applied K-medoid clustering algorithm to group semantically similar reply sentences in order to reduce overlapping reply sentences and at the same time generate distinct reply sentences in the summary. Finally, we used different quality text features to rank reply sentences in different clusters, and then the high-ranked reply sentences are selected from all clusters. The top ranked reply sentences from different clusters are combined to form a summary of discussion thread.

Our key contributions are given as follows:

(i) To develop a sentence embedding based clustering approach integrated with quality text features for discussion thread summarization.

(ii) To measure the efficacy of the suggested approach on two standard discussion forum datasets using ROUGE-N ($N = 1, 2$) evaluation metrics.

The paper is structured as follows: Related work is demonstrated in Section 2. Section 3 discusses the proposed summarization approach. The empirical results followed by discussion are described in Section 4. The conclusion, together with future work, is finally given in Section 5.

2. Related Work

First, we discuss the prior works on extractive summarization methods and then discuss the previous research efforts attempted for discussion thread summarization.

The methods used for summarizing the text can be alienated into two classes: extractive summarization (ES) and abstractive summarization (AS). ES aims to retrieve the most significant text units from the source document and combines them to create a condensed form of the document. On the other hand, AS is a challenging task and it requires deep semantic representation and advance natural language processing (NLP) techniques to produce a shorter novel text summary.

In extractive summarization, identification of the most significant textual units can be considered as a ranking problem, a classification problem, or a selection problem [6].

In selection approach [7, 8], the textual units are chosen individually in decreasing order of their importance/relevance, and at the same time both importance and redundancy of previously chosen textual units are taken into account. The classification approach [8, 9] treats each text unit independently and classifies it as either salient or nonsalient. The ranking approach presented in [10–17] assigns a salience score to each textual unit; then the textual units are arranged in decreasing order of salience scores, and the high significant units are chosen based on certain threshold or predefined cutoff (such as a defined number of text units/words).

For numerous document categories, the summarization units are typically sentences [18]. The units are usually utterances in meeting conversations summarization [19–22], and the typical units are posts in case of discussion thread summarization [23, 24].

There are many application domains where ES techniques are used, for instance, summaries of websites, patents records, and news stories [25, 26]. Tseng et al. [27] proposed a feature based summarization method for patents and used several features like cue phrases and sentence location to identify the relevant sentences. Trappey et al. [28] combined ontology tree structure and TF-IDF techniques to extract the keywords for selection of salient sentences from patent documents; then clustering technique is applied to group salient sentences to form a summary. Vazhenin et al. [29] provided Google's search engine an extended query based on WordNet to find the relevant webpages, and then the summary is created by choosing sentences form webpages having the most relevant keywords. Kallimani et al. [30] introduced a statistical method for news summarization. Different features such as term frequency, length of sentence, title of news article, proper nouns, and first sentence of news article are utilized to rank sentences in the news documents, and the high-ranked sentences are combined to create a summary. A pattern-based summarization method is introduced by [31] to score sentences in news documents by adding weights of the covered patterns.

Graph-based approaches [32], in the recent past, have also gained attraction and have been effectively utilized for extractive summarization. These approaches use either the

PageRank (PR) algorithm [33] or its other variations to rank graph nodes that represent different text units such as sentences/passages. The idea of connectivity graph presented by [34] assumes that the nodes having more connections with other nodes in a graph will carry more salient information. The Lex-PageRank technique [35] based on the idea of eigenvector centrality, produced a connectivity matrix of sentences and used PR-like algorithm to rank relevant sentences for summary. Another variation of PR [36] is also used for ranking significant sentences for summary. van Oortmerssen [24] examined subtopic features for many documents and embedded these features in graph-based ranking procedure. Affinity-graph approach [37] for extractive summarization used PR-like algorithm to calculate the relevance score of sentences by taking into consideration their information richness. A graph-based model for summarization presented by [38] thoroughly analyzed the document set information and examined its global impact at sentence level. A summarization approach based on weighted graph model [39] merged clustering and ranking methods for selection of relevant sentences. Nguyen-Hoang et al. [40] employed graph-based PR algorithm for summarization of Vietnamese documents. An extractive approach based on event graph [41] used human crafted rules for generation of multidocument summary.

Recently, the emergence of deep learning (DL) and reinforcement learning (RL) approaches [42–45] has gained attention of researchers, and their capabilities are exploited to enhance the text summarization task. However, the networks based on DL/RL need training on large amount of human crafted summaries, which are not easily available.

Discussion thread summarization (DTS) has been an exciting area of research, and some research efforts have been made in this direction over the last decade [24, 46, 47]. The goal of DTS is to select the most relevant reply posts in a discussion thread and merge them to form a concise thread summary. Most of the current work focused on comment threads summarization on news websites [48–53]. Previous research studies have effectively attempted abstractive summarization (AS) techniques for the task of DTS, such as e-mail threads summarization [3], summarizing written and spoken chats/conversations and meeting recordings [4], and summarizing Twitter topic [5].

In this study, we choose to propose extractive approach for discussion thread summarization. Individual replies of the forum users are assumed to be not rephrased in a different way during the retrieval of discussion threads.

Our work is a bit related to the works presented by the authors of [54, 55] who proposed a technique that combined topic modeling with clustering to produce a summary from forum posts. The technique was assessed on DUC 2007 standard dataset (for multidocument summarization) and the private discussion forum data. The evaluation results of the technique were compared with MEAD (a centroid-based summarization approach) [56]. The technique performed better than MEAD on DUC 2007; however, its performance was not consistently improved on forum data [54]. Bhatia et al. [55] treated discussion thread summarization (DTS) as a postclassification task, where the job is to classify a given forum post as either relevant to the summary or not. The classification was performed in a supervised manner as several features were used. The method was assessed on two standard forum datasets: Ubuntu and New York City (NYC). Ren [48] introduced a forum summarization technique that modelled the structures of forum replies in a discussion thread. The next section demonstrates the proposed methodology of our approach.

3. Proposed Methodology

The research framework of our proposed study is illustrated in Figure 1. It is composed of six phases: (1) preprocessing, (2) reply sentence embedding, (3) semantic clustering of replies, (4) text features extraction, (5) ranking of reply sentences, and (6) summary generation.

3.1. Preprocessing. The most significant procedure in computational linguistic/text summarization is data preprocessing. As the proposed work is related to discussion thread summarization, the preprocessing of thread documents is needed to speed up the subsequent computational steps. The preprocessing phase includes four steps, which are discussed as follows:

(a) Sentence segmentation: this step split the text into sentences by detecting boundary within text. Generally, an interrogation sign (?), sign of exclamation (!), or full stop/period (.) is used to indicate the sentence boundary [57].

Consider the following snippet of thread document: "I like Ubuntu. It is amongst the best flavors of Linux."

After segmenting the thread document, we get two sentences.

Input forum text:

"I like Ubuntu. It is amongst the best flavors of Linux."

Output:

Segment 1: "I like Ubuntu."

Segment 2: "It is amongst the best flavors of Linux."

(b) Tokenization: it is the procedure of segmenting sentences into distinct tokens (or words). Different whitespaces such as tabs, blanks, and punctuation symbols such as comma, semicolon, period, and colon are used as main cues for dividing the text into words.

(c) Stop words elimination: these are the words which exist in the thread document with high frequency. Stop words include prepositions, articles, conjunctions, and frequently occurring words like "an," "the," "a," "I," and so forth. These words convey minute or no meaning in the forum thread document, so elimination of stop words from the thread document will assist in boosting the system performance. This work used a list of stop words proposed by Buckley [58].

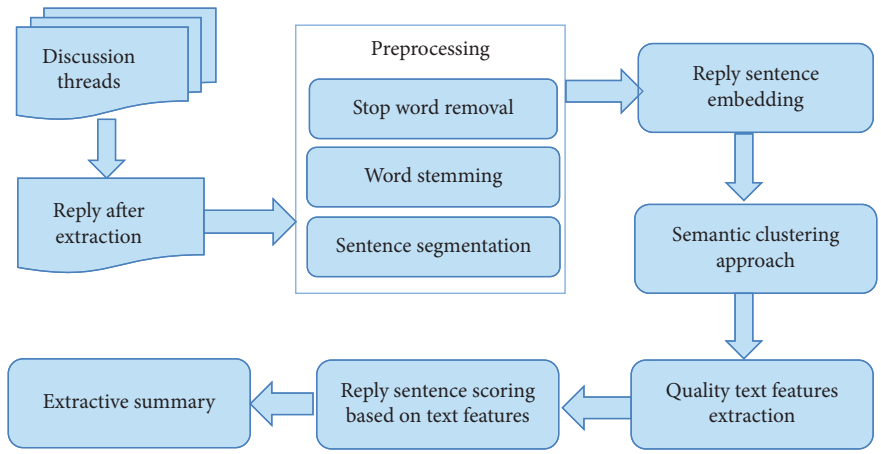

Figure 1: Proposed approach for discussion thread summarization.

(d) Word stemming: it is one of the significant tasks of preprocessing phase. It converts the derived terms to its root term for grasping the similar notion. This study used a renowned algorithm called Porter's stemming [58], which eliminates suffixes from the derived words. For instance, the stemming algorithm will convert the words "playing," "played," and "plays" to their stem word "play" by removing suffixes -ing, -ed, and -s.

3.2. Reply Sentence Embedding. The purpose of this step is to get sentence embeddings from the thread replies sentences. Sentence embedding is a numerical representation of text and is efficiently used by machine learning algorithm. It is a richer representation of text which preserves both semantic and syntactic information in sentences and leads to enhanced performance in almost each NLP task.

We used word2vec model to get sentence embeddings from the collection of reply sentences by extracting word vectors/embeddings for each individual word/token in the reply sentences. This study used the pretrained word2vec model [59, 60], released by Google, to learn word vectors (word embeddings) for each word in the reply sentences. The trained word2vec model is based on neural network architecture and employs a continuous Bag of Words to learn distributed vector representations of words. It is trained roughly on one hundred billion words present in Google News dataset. We set length of word vector to the default 300 features.

Finally, sentence embeddings/sentences vectors are obtained by using the average of all word vectors that are present in word2vec vocabulary and ignoring the missing word vectors. Each reply sentence is now expressed as a numeric vector. We also stored the sentence text along with its numeric representation for later use in sentence selection for summaries based on different text features.

3.3. Semantic Clustering of Reply Sentences. Usually, different users in a discussion thread apparently post different replies to the initial question post but they carry the same meanings; so it is a good idea to cluster semantically similar replies in a thread. In this study, we employed an unsupervised machine learning algorithm called K-medoid to group semantically similar reply sentences in order to reduce overlapping reply sentences and at the same time generate distinct reply sentences in the summary.

The K-medoid algorithm [61] is a partitioning clustering approach that separates the dataset of n data points (reply sentences) into K predefined distinctive nonoverlapping groups called clusters, where each data point (reply sentence) goes into one cluster. K-medoid clustering is more stable and less susceptible to outliers and noise in comparison to K-means clustering algorithm, since it uses medoids (actual points) as cluster centers instead of average of points used in K-means. Moreover, it is fast and converges in a specific number of steps; it is simple and easy to implement. In this clustering approach, each cluster is characterized by one of the data points in cluster, and such data points are called cluster medoids.

K-medoid is also known as partitioning around medoid (PAM). The term medoid refers to the point in the cluster whose average dissimilarity with all other points in the cluster is minimal. The key objective of K-medoid clustering is to diminish the summation of dissimilarities between data points in a given cluster and the respective cluster medoid (cluster center). The cost of K-medoid algorithm [61] is given as

$$C = \sum_{P_i \in C_i} \sum_{P_i \in C_i} |P_i - C_i|. \quad (1)$$

The pseudocode of K-medoid clustering algorithm is as follows:

(1) Initialization: Select medoids by randomly selecting k points from a set of n data points.

(2) Link each data point to the nearby medoid by using Manhattan distance.

(3) While the cost reduces, for every medoid c, for every data point p not selected as medoid:

(a) Exchange c and p, link each data point to the nearest medoid, compute the cost again.
(b) Compare total cost with the ones in the previous step; if it is greater then undo the exchange.

In this work, we selected $k = 10$ to be the optimum number of clusters calculated using silhouette method. The number of sentences that will form the final summary is dependent on the optimum number of clusters. Here the number of selected clusters is ten, and we choose top scored representative, salient, and information-rich sentences from ten clusters based on different text features to yield a summary.

3.4. Quality Text Features Extraction. Text features play important role in choosing salient content for summary generation. In order to score reply sentences in different clusters, we chose eight different quality text features as discussed below. This step extracts 8 different quality text features from reply sentences in each cluster formed in previous step. The feature values are normalized between 0 and 1. Next, we briefly discuss the text features used in this work.

3.4.1. Semantic Distance between Thread Reply and Thread Centroid. This feature computes the similarity between thread reply sentence and thread centroid. The reply sentences that are semantically similar to thread centroid are believed to be appropriate for the final summary. We employed TF-IDF technique to determine the thread centroid, which represents the most important features/words in a thread. Once the thread centroid is computed for a given discussion thread, then we determine the semantic distance between thread centroid and thread replies by using a technique called word mover's distance (WMD). WMD is a word embedding technique that uses Google's pretrained word2vec model to get similarity between thread reply and thread centroid.

$$\text{reply\_sent}_{f1} = \text{WMD}(\text{reply sentence}, \text{thread centroid}). \tag{2}$$

3.4.2. Cosine Similarity between Reply and Thread Centroid. Reply sentences closely related to the thread centroid are assumed to be important for inclusion in summary. This feature finds the cosine similarity between vector representations of thread centroid and thread reply sentences.

$$\text{reply\_sent}_{f2} = \text{Cosine\_Sim}(\text{reply sentence}, \text{thread centroid}). \tag{3}$$

3.4.3. Unique Words Count in a Reply. This feature finds the number of unique words in a thread reply sentence. The reply sentences with unique words are considered appropriate for summary.

$$\text{reply\_sent}_{f3} = \text{Unique\_words}(\text{reply sentence}). \tag{4}$$

3.4.4. Common or Overlapping Words between Thread Reply and Initial Post. A thread reply sentence is important for summary if it has matching words in the initial post. This feature determines the number of common or overlapping words between thread reply and initial post using Jaccard similarity.

$$\text{reply\_sent}_{f4} = \text{Jaccard\_Sim}(\text{reply sentence}, \text{initial post}). \tag{5}$$

3.4.5. Semantic Distance between Thread Reply and Thread Title. A thread reply sentence that is semantically similar to the thread title is considered to be significant for summary. We used WMD to determine the semantic distance between thread title and thread reply sentence.

$$\text{reply\_sent}_{f5} = \text{WMD}(\text{reply sentence}, \text{thread title}). \tag{6}$$

3.4.6. Semantic Distance between Thread Reply and Initial Post. A thread reply sentence that is semantically similar to the initial post is considered to be salient for summary. WMD is employed to get the semantic gap/distance between thread reply sentence and initial post.

$$\text{reply\_sent}_{f6} = \text{WMD}(\text{reply sentence}, \text{initial post}). \tag{7}$$

3.4.7. Reply Sentence Length. It determines the length of the reply sentence by finding the number of words in it.

$$\text{reply\_sent}_{f7} = \frac{\text{No. of words in reply sentence}}{\text{Max length of reply sentence}}. \tag{8}$$

3.4.8. Number of Verbs and Nouns. This feature determines the number of verbs and nouns in a thread reply.

$$\text{reply\_sent}_{f8} = \frac{\text{No. of verbs and nouns in reply sentence}}{\text{Reply sentence length}}. \tag{9}$$

3.5. Ranking of Reply Sentences. This objective of this step is to select the best scored reply sentences from the diverse clusters based on different quality text features as discussed in Section 3.4. Each reply sentence in a cluster is represented by a vector of 8 dimensions, whereas each dimension represents the feature score.

$$\begin{aligned}\text{reply\_sent} = \big[&\text{reply\_sent}_{f_1}, \text{reply\_sent}_{f_2}, \text{reply\_sent}_{f_3}, \text{reply\_sent}_{f_4}, \text{reply\_sent}_{f_5},\\ &\text{reply\_sent}_{f_6}, \text{reply\_sent}_{f_7}, \text{reply\_sent}_{f_8}\big].\end{aligned} \tag{10}$$

Once the features scores for reply sentences in different clusters are calculated, these features scores are summed up to get a ranking score for each reply sentence as given in the following equation:

$$\text{Score}(\text{reply\_sent}) = \sum_{k=1}^{8} \text{reply\_sent}_{f_i}, \tag{11}$$

where Score(reply_sent) represents the ranking score of reply sentence and reply_sent_{f_i} indicates the reply sentence features score. Once the score of reply sentences is obtained using equation (11), the reply sentences are ranked in different clusters based on these scores, and the top scored sentences are picked from all clusters. In this study, the number of selected clusters is 10, so we choose ten representative reply sentences from 10 clusters for final summary generation.

3.6. Summary Generation. In this step, the reply sentences with maximum rank score are chosen from each cluster as representative sentences to form an extractive summary of discussion thread. The rank score for each reply sentence within cluster is obtained based on different quality text features that are discussed in previous section.

4. Experimental Settings

4.1. Datasets for Evaluation. Our sentence embedding based clustering approach for discussion thread summarization and other state-of-the-art clustering techniques are evaluated on two publicly available discussion forum datasets—technical discussion forum for Ubuntu Linux distribution (http://ubuntuforums.org) and nontechnical discussion forum for New York City (NYC) called online TripAdvisor (https://www.tripadvisor.com.my/ShowForum-g28953-i4-New_York.html). Hundred discussion threads were randomly chosen from both datasets. Each thread has initial post called question and associated replies as candidate answers. There are a total of 756 replies in Ubuntu dataset and 788 replies in NYC dataset.

For each discussion thread in both Ubuntu and NYC datasets, there are also associated Gold summaries created by 2 human annotators named as Annotator-1 and Annotator-2. The effectiveness of the proposed method is assessed using ROUGE-N ($N = 1, 2$) evaluation metrics.

4.2. Experimental Steps. Given the corpus of discussion threads, at first, the preprocessing techniques are applied to divide the corpus into sentences, split the sentences into tokens (words), and eliminate the stop words. Then, the procedure of porter stemming is applied to the remaining words in order to transform them into their root words. Next, word2vec model is utilized to transform the collection of reply sentences into sentence embeddings/sentence vectors by extracting word embedding for each word in the reply sentences. We used the pretrained word2vec model to learn word vectors for each word in all reply sentences. We set the length of word vector to the default 300 features. Finally, sentence vectors/embeddings are obtained by using the average of all word vectors in reply sentences. Next, we employed K-medoid clustering algorithm to group semantically similar reply sentences in order to reduce overlapping reply sentences. We chose 10 optimal clusters in this work. Different quality text features are used to rank reply sentences in different clusters, and sentences with high ranks are picked from all clusters. The top ten representative reply sentences from 10 different clusters formed the final extractive summary.

In order to assess the performance of the proposed sentence embedding based clustering approach for discussion thread summarization, we set up two comparison models for summarization task. The first model is fuzzy c-means clustering (FCM) [61], which attempts to divide a finite collection of n data points into a collection of K clusters by linking each data point with all clusters through a real valued vector of indexes. Unlike traditional clustering, each data point in fuzzy clustering goes to one or more clusters at the same time.

The second model is K-means (KM) clustering [62], which attempts to divide the dataset into K predefined distinctive nonoverlapping groups called clusters, whereas each data point is allocated to a single cluster. The data points in our case refer to reply sentences (represented as sentence embeddings/vectors). The key objective of the KM algorithm is to reduce the sum of semantic gaps/distances between data points and their respective cluster centroids.

This research uses ROUGE-N ($N = 1, 2$) assessment metrics to compare the efficiency of suggested summarization method with other comparison models.

ROUGE evaluation metric is effectively applied in the field of extractive summarization task [63]. ROUGE-N describes an intersection of n-grams between the system generated summary and human annotator (reference) summary and is determined using the following equation:

$$\text{ROUGE} - N = \frac{\sum_{S \in \{\text{refrence\_summaries}\}} \sum_{\text{gram}_n \in S} \text{Count}_{\text{match}}(\text{gram}_n)}{\sum_{S \in \{\text{refrence\_summaries}\}} \sum_{\text{gram}_n \in S} \text{Count}(\text{gram}_n)}, \tag{12}$$

where $\text{count}_{\text{match}}(\text{gram}_n)$ is the highest n-gram, which exists at the same time in both system (machine) summary and human (annotator) summaries, and gram_n is length of n-gram.

The different measures for system summary [63] are determined as follows:

$$\text{Precision} = \frac{\text{System Summary} \cap \text{Human Summary}}{\text{System Summary}},$$

$$\text{Recall} = \frac{\text{System Summary} \cap \text{Human Summary}}{\text{Human Summary}},$$

$$F - \text{Measure} = 2 * \frac{\text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}}. \tag{13}$$

Tables 1 and 2 illustrate the outcomes of comparative assessment of the proposed summarization approach and

Table 1: Evaluation of summarization models on various measures using *ROUGE-1* on Ubuntu dataset.

Summarization models	Avg-precision	Avg-recall	Avg-*F*-measure
Proposed approach	**34.68**	**39.83**	**36.03**
Fuzzy c-means	28.20	28.31	27.46
K-means	25.41	27.52	26.11

Table 2: Evaluation of summarization models on various measures using *ROUGE-2* on Ubuntu dataset.

Summarization models	Avg-precision	Avg-recall	Avg-*F*-measure
Proposed approach	**11.53**	**11.44**	**11.41**
Fuzzy c-means	7.23	7.02	7.00
K-means	6.44	6.98	6.51

other approaches determined using ROUGE-N ($N = 1, 2$) measures. These outcomes are obtained from a subset of 100 randomly chosen discussion threads from Ubuntu dataset. Considering the outcomes of ROUGE-1 shown in Table 1, the proposed sentence embedding based clustering approach for discussion thread summarization works better than other clustering techniques based on average precision, *F*-measure, and recall. On the other hand, FCM clustering approach generates better summarization results than KM clustering approach.

Likewise, considering the ROUGE-2 findings shown in Table 2, the proposed summarization method also outperforms other clustering-based techniques based on different measures. FCM clustering was sustained to produce improved summarization outcomes compared to KM clustering.

For Ubuntu dataset, Figures 2 and 3 illustrate the summarization outcomes of the suggested approach and other summarization approaches, calculated using ROUGE-N ($N = 1, 2$) metrics.

Similarly, for NYC dataset, Tables 3 and 4 show the outcomes of comparative assessment of the proposed summarization method and other comparison models using ROUGE-N ($N = 1, 2$) measures. These results are also obtained from a subset of 100 randomly chosen discussion threads from NYC dataset.

Considering the ROUGE-1 outcomes shown in Table 3, the proposed clustering technique gives superior summarization results compared to other clustering techniques based on average recall, precision, and *F*-measure. On the other hand, FCM clustering yields better results than KM algorithm in terms of average precision; however, it yields lower average recall and average *F*-measure than KM algorithm.

The ROUGE-2 results given in Table 4 show that proposed sentence embedding based clustering approach for discussion thread summarization is still stable and performs better than other clustering techniques. However, the performance of KM algorithm for summarization task on NYC dataset is better than FCM algorithm. For NYC dataset, Figures 4 and 5 depict the summarization outcomes of the proposed method and other related summarization models, determined using ROUGE-N ($N = 1, 2$) metrics.

4.3. Discussion. This section illustrates the thread summarization outcomes of our proposed sentence embedding based clustering approach and other comparison summarization approaches in the context of Ubuntu and NYC discussion forums datasets. The efficacy of the suggested approach and other summarization approaches is measured in terms of average recall, precision, and *F*-measure achieved with ROUGE-N ($N = 1, 2$) metrics.

Referring to the outcomes of ROUGE-1 shown in Tables 1 and 3, the proposed approach gave better summarization results in comparison to FCM and KM clustering algorithms in the context of precision, recall, and *F*-measure. On the other hand, FCM gave better summarization results than KM algorithm on Ubuntu dataset. However, KM algorithm showed better performance than FCM in terms of average recall and average *F*-measure on NYC dataset.

It can be observed from the ROUGE-2 results given in Tables 2 and 4 that the proposed approach showed stable performance and gave superior summarization results than other clustering techniques. On the other hand, ROUGE-2 results of K-means algorithm are better than fuzzy c-means clustering algorithm. Experimental results support that proposed sentence embedding based clustering approach showed stable and improved performance on both Ubuntu and NYC datasets as compared to other clustering techniques. In essence, it can also be observed that sentence embedding based clustering approach combined with ranking procedure based on quality text features boosted the summarization results.

We also conducted statistical *T*-tests to validate the empirical results in order to reveal the enhancement of our suggested approach over other summarization models. The paired-samples *T*-test procedure is used in this work to find the mean difference of two outcomes that express the same test set and got tiny significance values of 0.032, 0.027, and 0.025 for average precision, recall, and *F*-measure, respectively. The tiny significance values produced for the *T*-test

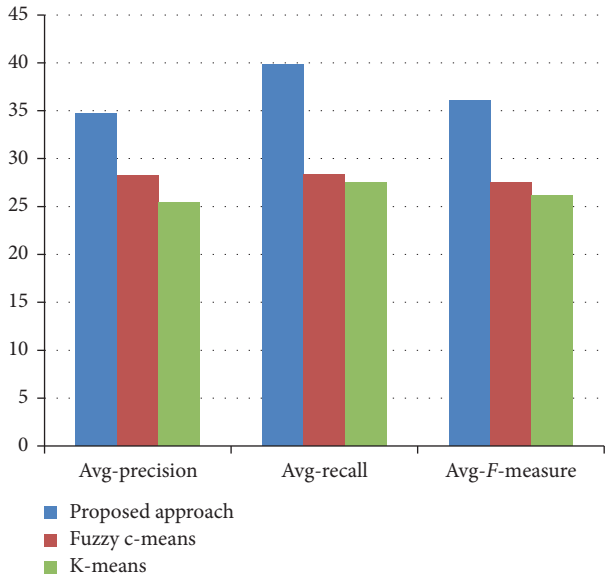

Figure 2: Evaluation of summarization models in respect of ROUGE-1 measures on Ubuntu dataset.

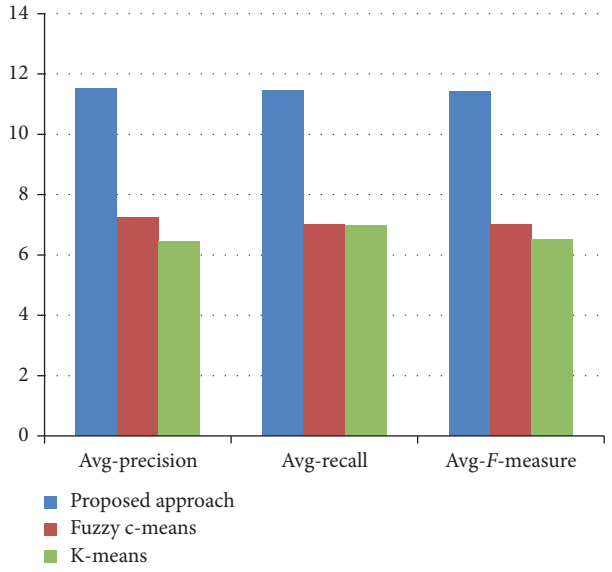

Figure 3: Evaluation of summarization models in respect of ROUGE-2 measures on Ubuntu dataset.

Table 3: Evaluation of summarization models on various measures using *ROUGE*-1 on NYC dataset.

Summarization models	Avg-precision	Avg-recall	Avg-*F*-measure
Proposed approach	**31.20**	**37.96**	**31.80**
Fuzzy c-means	24.96	28.71	25.14
K-means	24.55	29.66	27.09

Table 4: Evaluation of summarization models on various measures using *ROUGE* -2 on NYC dataset.

Summarization models	Avg-precision	Avg-recall	Avg-*F*-measure
Proposed approach	**8.27**	**10.94**	**9.18**
Fuzzy c-means	4.77	5.38	5.00
K-means	5.54	7.34	6.10

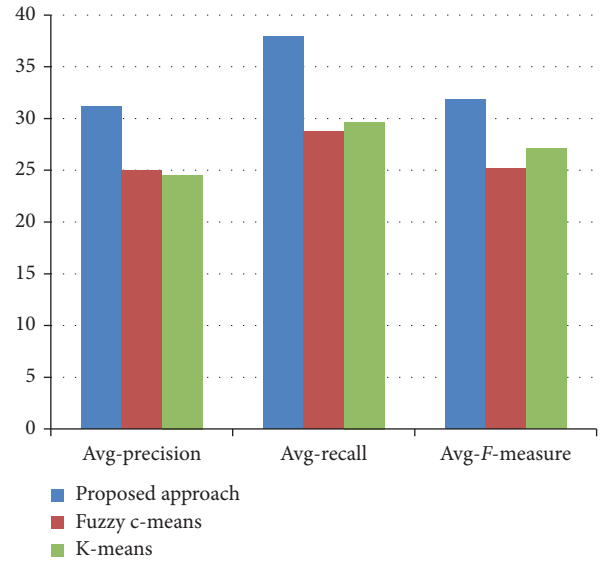

Figure 4: Evaluation of summarization models on different measures using *ROUGE* -1 on NYC dataset.

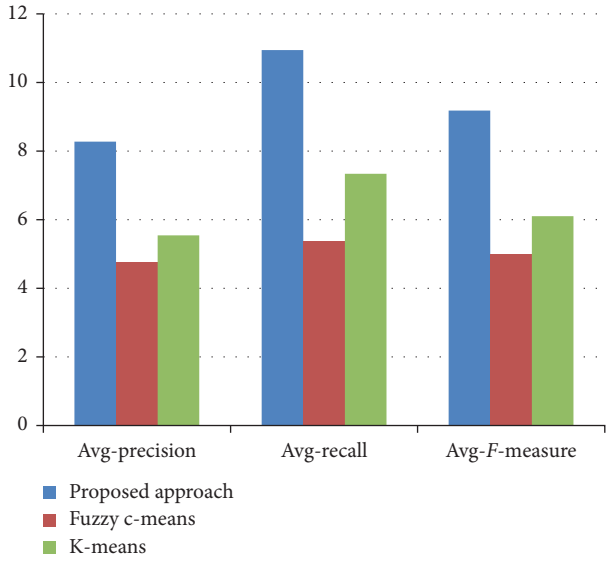

Figure 5: Evaluation of summarization models on different measures using *ROUGE* -2 on NYC dataset.

(usually < 0.05) illustrate that the outcomes of the suggested approach and summarization models to be compared are significantly different.

5. Conclusion and Future Work

Discussion thread summarization is a daunting task, and this work sets a viable direction for thread summarization task. We introduced a sentence embedding based clustering approach that takes semantic representation of thread replies, groups semantically similar replies in different clusters, and then creates extractive summary by selecting the top ranked replies from different cluster based on various quality text features. The summary gives a gist of enormous amount of thread replies. The proposed approach is fully automatic and generic and is appropriate for discussion forums from different domains. From the experimental findings, we have confirmed that the proposed method has produced better results compared to other summarization methods. In the future, we are planning to use deep learning models to create extractive/abstractive summary of the discussion threads. In addition, we expand our methodology to other domains and inspect the usefulness of the proposed methodology.

Data Availability

The data are publicly available at https://ubuntuforums.org and https://www.tripadvisor.com.my/ShowForum-g28953-i4-New_York.html.

Conflicts of Interest

The authors declare no conflicts of interest.

Acknowledgments

This work was supported by Islamia College, Peshawar, and Higher Education Commission (HEC) of Pakistan.

References

[1] L. Z. a. E. Hovy, "On the summarization of dynamically introduced information discussion and blogs," in *Proceedings of the 2006 AAAI Spring Symposium*, Stanford, CA, USA, 2006.

[2] A. Osman, N. Salim, and F. Saeed, "Quality dimensions features for identifying high-quality user replies in text forum threads using classification methods," *PLoS One*, vol. 14, no. 5, Article ID e0215516, 2019.

[3] D. M. Zajic, B. J. Dorr, and J. Lin, "Single-document and multi-document summarization techniques for email threads using sentence compression," *Information Processing & Management*, vol. 44, no. 4, pp. 1600–1610, 2008.

[4] Y. Mehdad, G. Carenini, and R. Ng, "Abstractive summarization of spoken and written conversations based on phrasal queries," in *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, Baltimore, MD, USA, 2014.

[5] R. Zhang, W. Li, D. Gao, and Y. Ouyang, "Automatic twitter topic summarization with speech acts," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 21, no. 3, pp. 649–658, 2012.

[6] D. Das and A. Martins, "A survey on automatic text summarization," *Literature Survey for Language and Statistics. II Course at CMU*, Carnegie Mellon University, Pittsburgh, PA, USA, 2007.

[7] J. Carbonell and J. Goldstein, "The use of MMR, diversity-based reranking for reordering documents and producing summaries," in *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, Melbourne Australia, August 1998.

[8] R. Nallapati, F. Zhai, and B. Z. Summarunner, "A recurrent neural network based sequence model for extractive summarization of documents," in *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, San Francisco, CA, USA, 2017.

[9] J. Kupiec, J. Pedersen, and F. Chen, "A trainable document summarizer," in *Proceedings of the 18th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, Seattle WC, USA, 1995.

[10] Y. Gong and X. Liu, "Generic text summarization using relevance measure and latent semantic analysis," in *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, New Orleans, LA, USA, 2001.

[11] K. Svore, L. Vanderwende, and C. Burges, "Enhancing single-document summarization by combining RankNet and third-party sources," in *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, Prague, Czech Republic, 2007.

[12] K. Toutanova, "The pythy summarization system: Microsoft research at DUC 2007," in *Proceedings of the DUC*, New York, NY, USA, 2007.

[13] D. Metzler and T. Kanungo, "Machine learned sentence selection strategies for query-biased summarization," in *Proceedings of the Sigir Learning to Rank Workshop*, Singapore, 2008.

[14] M. R. Amini and N. Usunier, "Incorporating prior knowledge into a transductive ranking algorithm for multi-document summarization," in *Proceedings of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, Boston, MA, USA, 2009.

[15] C. Shen and T. Li, "Learning to rank for query-focused multi-document summarization," in *Proceedings of the 2011 IEEE 11th International Conference on Data Mining*, IEEE, Vancouver, CanadaIEEE, Vancouver, Canada, 2011.

[16] R. Sipos, P. Shivaswamy, and T. Joachims, "Large-margin learning of submodular summarization models," in *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, Association for Computational Linguistics, Avignon, FranceAssociation for Computational Linguistics, Avignon, France, 2012.

[17] A. Dlikman and M. Last, "Using machine learning methods and linguistic features in single-document extractive summarization," in *Proceedings of the DMNLP@PKDD/ECML*, Riva del Garda, Italy, 2016.

[18] V. Gupta and G. S. Lehal, "A survey of text summarization extractive techniques," *Journal of Emerging Technologies in Web Intelligence*, vol. 2, no. 3, pp. 258–268, 2010.

[19] G. Murray, S. Renals, and J. Carletta, "Extractive summarization of meeting recordings," in *Proceedings of the 9th European Conference on Speech Communication and Technology*, Lisbon, Portugal, September 2005.

[20] F. Liu and Y. Liu, "Correlation between rouge and human evaluation of extractive meeting summaries," in *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics on Human Language Technologies*, Association for Computational Linguistics, Columbus, OH, USAAssociation for Computational Linguistics, Columbus, OH, USA, 2008.

[21] G. Penn and X. Zhu, "A critical reassessment of evaluation baselines for speech summarization," in *Proceedings of the ACL-08*, HLT, Columbus, OH, USAHLT, Columbus, OH, USA, 2008.

[22] M. Marge, S. Banerjee, and A. I. Rudnicky, "Using the Amazon mechanical turk to transcribe and annotate meeting speech for extractive summarization," in *Proceedings of the NAACL HLT 2010 Workshop on Creating Speech and Language Data with Amazon's Mechanical Turk*, Association for Computational Linguistics, Los Angeles, CA, USAAssociation for Computational Linguistics, Los Angeles, CA, USA, 2010.

[23] S. Bhatia and P. Mitra, "Adopting inference networks for online thread retrieval," in *Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence*, Atlanta, GA, USA, 2010.

[24] G. van Oortmerssen, "Analyzing cancer forum discussions with text mining," in *Proceedings of the 2nd International Workshop on Extraction and Processing of Rich Semantics from Medical Texts*, vol. 127, Tokyo, Japan, 2017.

[25] H. Jeong, Y. Ko, and J. Seo, "How to improve text summarization and classification by mutual cooperation on an integrated framework," *Expert Systems with Applications*, vol. 60, pp. 222–233, 2016.

[26] C. N. Silla Jr., C. A. Kaestner, and A. A. Freitas, "A non-linear topic detection method for text summarization using wordnet," in *Proceedins of the Workshop of Technology Information Language Human (TIL'2003)*, São Carlos, Brazil, 2003.

[27] Y.-H. Tseng, Y.-M. Wang, Y.-I. Lin, C.-J. Lin, and D.-W. Juang, "Patent surrogate extraction and evaluation in the context of patent mapping," *Journal of Information Science*, vol. 33, no. 6, pp. 718–736, 2007.

[28] A. J. C. Trappey, C. V. Trappey, and C.-Y. Wu, "Automatic patent document summarization for collaborative knowledge systems and services," *Journal of Systems Science and Systems Engineering*, vol. 18, no. 1, pp. 71–94, 2009.

[29] D. Vazhenin, S. Ishikawa, and V. Klyuev, "A user-oriented web retrieval summarization tool," in *Proceedings of the 2009 Second International Conference on Advances in Human-Oriented and Personalized Mechanisms, Technologies, and Services*, IEEE, Porto, PortugalIEEE, Porto, Portugal, 2009.

[30] J. S. Kallimani, K. G. Srinivasa, and B. Eswara Reddy, "Summarizing news paper articles: experiments with ontology-based, customized, extractive text summary and word scoring," *Cybernetics and Information Technologies*, vol. 12, no. 2, pp. 34–50, 2012.

[31] J.-P. Qiang, P. Chen, W. Ding, F. Xie, and X. Wu, "Multi-document summarization using closed patterns," *Knowledge-Based Systems*, vol. 99, pp. 28–38, 2016.

[32] X. Han, "Text summarization using framenet-based semantic graph model," *Scientific Programming*, vol. 2016, Article ID 5130603, 10 pages, 2016.

[33] L. Page, *The PageRank Citation Ranking: Bringing Order to the Web*, Stanford University, Stanford, CA, USA, 1999.

[34] I. Mani and E. Bloedorn, "Summarizing similarities and differences among related documents," *Information Retrieval*, vol. 1, no. 1-2, pp. 35–67, 1999.

[35] G. Erkan and D. R. Radev, "LexPageRank: prestige in multi-document text summarization," in *Proceedings of the EMNLP 2004*, ACL, Barcelona, SpainACL, Barcelona, Spain, 2004.

[36] R. Mihalcea and P. Tarau, "A language independent algorithm for single and multiple document summarization," in *Proceedings of the IJCNLP'2005*, Jeju Island, Korea, 2005.

[37] X. Wan and J. Yang, "Improved affinity graph based multi-document summarization," in *Proceedings of the Human Language Technology Conference of the NAACL*, ACL, New York, NY, USAACL, New York, NY, USA, 2006.

[38] F. Wei, W. Li, Q. Lu, and Y. He, "A document-sensitive graph model for multi-document summarization," *Knowledge and Information Systems*, vol. 22, no. 2, pp. 245–259, 2010.

[39] S. S. Ge, Z. Zhang, and H. He, "Weighted graph model based sentence clustering and ranking for document summarization," in *Proceedings of the 4th International Conference on*

Interaction Sciences, IEEE, Busan, South KoreaIEEE, Busan, South Korea, 2011.

[40] T.-A. Nguyen-Hoang, K. Nguyen, and Q.-V. Tran, "TSGVi: a graph-based summarization system for Vietnamese documents," *Journal of Ambient Intelligence and Humanized Computing*, vol. 3, no. 4, pp. 305–313, 2012.

[41] G. Glavaš and J. Šnajder, "Event graphs for information retrieval and multi-document summarization," *Expert Systems with Applications*, vol. 41, no. 15, pp. 6904–6916, 2014.

[42] Y. Dong, "Banditsum: extractive summarization as a contextual bandit," 2018, https://arxiv.org/abs/1809.09672.

[43] Y. Liu, "Fine-tune BERT for extractive summarization," 2019, https://arxiv.org/abs/1903.10318.

[44] Y. Liu and M. Lapata, "Text summarization with pretrained encoders," 2019, https://arxiv.org/abs/1908.08345.

[45] A. Joshi, E. Fidalgo, E. Alegre, and L. Fernández-Robles, "SummCoder: an unsupervised framework for extractive text summarization based on deep auto-encoders," *Expert Systems with Applications*, vol. 129, pp. 200–215, 2019.

[46] L. Zhou and E. H. Hovy, "On the summarization of dynamically introduced information: online discussions and blogs," in *Proceedings of the AAAI Spring Symposium: Computational Approaches to Analyzing Weblogs*, Boston, MA, USA, 2006.

[47] A. S. Tigelaar, R. Op Den Akker, and D. Hiemstra, "Automatic summarisation of discussion fora," *Natural Language Engineering*, vol. 16, no. 2, pp. 161–192, 2010.

[48] Z. Ren, "Summarizing web forum threads based on a latent topic propagation process," in *Proceedings of the 20th ACM International Conference on Information and Knowledge Management*, Scotland, UK, 2011.

[49] C. Llewellyn, C. Grover, and J. Oberlander, "Summarizing newspaper comments," in *Proceedings of the Eighth International AAAI Conference on Weblogs and Social Media*, Ann Arbor, MI, USA, 2014.

[50] G. Giannakopoulos, "Multiling 2015: multilingual summarization of single and multi-documents, on-line, and call-center conversations," in *Proceedings of the 16th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, Prague, Czech Republic, 2015.

[51] M. Kabadjov, "Onforums: the shared task on online forum summarisation at multiling'15," in *Proceedings of the 7th Forum for Information Retrieval Evaluation*, Gandhinagar, India, 2015.

[52] A. Aker, "Automatic label generation for news comment clusters," in *Proceedings of the 9th International Natural Language Generation Conference*, Edinburgh, Scotland, 2016.

[53] E. Barker, "The SENSEI annotated corpus: human summaries of reader comment conversations in on-line news," in *Proceedings of the 17th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, Los Angeles, CA, USA, 2016.

[54] J. Krishnamani, Y. Zhao, and R. Sunderraman, "Forum summarization using topic models and content-metadata sensitive clustering," in *Proceedings of the 2013 IEEE/WIC/ACM International Joint Conferences on Web Intelligence (WI) and Intelligent Agent Technologies (IAT)*, IEEE, Atlanta, GeorgiaIEEE, Atlanta, Georgia, 2013.

[55] S. Bhatia, P. Biyani, and P. Mitra, "Summarizing online forum discussions–can dialog acts of individual messages help?" in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Doha, Qatar, 2014.

[56] G. Erkan and D. R. Radev, "Lexrank: graph-based lexical centrality as salience in text summarization," *Journal of Artificial Intelligence Research*, vol. 22, pp. 457–479, 2004.

[57] A. Mikheev, "Document centered approach to text normalization," in *Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, Athens, Greece, 2000.

[58] C. Buckley, *Automatic Query Expansion Using SMART: TREC 3*, NIST Special Publication, Gaithersburg, MD, USA, 1995.

[59] B. Pang and L. Lee, "A sentimental education: sentiment analysis using subjectivity summarization based on minimum cuts," in *Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics*, Association for Computational Linguistics, Barcelona, SpainAssociation for Computational Linguistics, Barcelona, Spain, 2004.

[60] A. L. Maas, "Learning word vectors for sentiment analysis," in *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, Association for Computational Linguistics, Portland, OR, USAAssociation for Computational Linguistics, Portland, OR, USA, 2011.

[61] J. C. Bezdek, R. Ehrlich, and W. Full, "FCM: the fuzzy c-means clustering algorithm," *Computers & Geosciences*, vol. 10, no. 2-3, pp. 191–203, 1984.

[62] T. Kanungo, D. M. Mount, N. S. Netanyahu, C. D. Piatko, R. Silverman, and A. Y. Wu, "An efficient k-means clustering algorithm: analysis and implementation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 7, pp. 881–892, 2002.

[63] C.-Y. Lin, "Rouge: a package for automatic evaluation of summaries," in *Proceedings of the ACL-04 Workshop*, ACL, Barcelona, SpainACL, Barcelona, Spain, 2004.