Effectiveness of Fuzzy Graph Based Document Model

Aswathy M R[1*], P.C. Reghu Raj[2], and Ajeesh Ramanujan[3]
[1]Research Scholar, APJ Kerala Technological University,
Govt.Engineering College, Palakkad, India
[e-mail: aswathy.moozhiyil@gmail.com]
APJ Kerala Technological University,
[2]Department of Computer Science & Engineering,
Govt. Engineering College, Kozhikode, India.
[e-mail: pcreghu@gmail.com]
APJ Kerala Technological University,
[3]College of Engineering, Trivandrum India.
[e-mail: ajeeshramanujan@gmail.com]
[*]Corresponding author: Aswathy M R

Received December 29, 2023; revised May 17, 2024; revised July 26, 2024; accepted August 4, 2024; published August 31, 2024

Abstract

Graph-based document models have good capabilities to reveal inter-dependencies among unstructured text data. Natural language processing (NLP) systems that use such models as an intermediate representation have shown good performance. This paper proposes a novel fuzzy graph-based document model and to demonstrate its effectiveness by applying fuzzy logic tools for text summarization. The proposed system accepts a text document as input and identifies some of its sentence level features, namely sentence position, sentence length, numerical data, thematic word, proper noun, title feature, upper case feature, and sentence similarity. The fuzzy membership value of each feature is computed from the sentences. We also propose a novel algorithm to construct the fuzzy graph as an intermediate representation of the input document. The Recall-Oriented Understudy for Gisting Evaluation (ROUGE) metric is used to evaluate the model. The evaluation based on different quality metrics was also performed to verify the effectiveness of the model. The ANOVA test confirms the hypothesis that the proposed model improves the summarizer performance by 10% when compared with the state-of-the-art summarizers employing alternate intermediate representations for the input text.

Keywords: Eigenvalue, Fuzzy graph, Membership function, Natural language processing, Summary generation, Text features.

1. Introduction

The performance of text processing applications depends on the goodness of the document representation used by them. Various text representation models commonly used in natural language processing (NLP) are Term Frequency-Inverse Document Frequency (TF-IDF), Bag of Words (BOW), word embeddings such as Word2Vec and Global Vectors for word representation (GloVe), and transformer-based models such as Bidirectional Encoder Representations from Transformers (BERT), each having own strengths and weaknesses in processing unstructured text. TF-IDF is useful for determining the importance of a word within a document based on its frequency and across multiple documents. But TF-IDF cannot identify the words even with a slight change or cannot check the semantics of the documents [1]. BOW represents text as a vector disregarding word order but limited in capturing semantic relationships [2]. Word embeddings provide a way to represent words as numerical vectors, enabling machines to understand the implicit meaning of words based on their context in the text. Models like Word2Vec [3] and GloVe [4] excel at capturing these semantic relationships. BERT [5] on the other hand, takes contextual information into consideration and uses neural networks for understanding words and phrases in a sentence.

Graph models offer a different perspective by representing entities as vertices and relationships as edge weights [6]. Generally, graphs do not accommodate varying degrees of connection strength easily, or may struggle to represent uncertainty or ambiguity in natural language effectively [7]. The proposed fuzzy graph document model aims to address this limitation by leveraging fuzzy logic principles to compute relationship degrees between connected entities more effectively.

The fuzzy graph model is constructed by extracting various sentence-level features from the input document. The model uses fuzzy membership functions and assigns edge weights in the graph. Moreover, it enhances the edge weights by resolving coreferences in the input document.

The paper is structured as follows: Section 2 surveys the related work on the fuzzy graph model, sentence ranking, and text summarization, Section 3 defines the problem, and describes the design and implementation of the proposed system. The evaluation strategy, results obtained, and the observations are covered in Section 4. Section 5 discusses the limitations of the work. Concluding remarks are offered in Section 6.

2. Related Work

In order to process unstructured text for various NLP tasks, an intermediate representation such as a vector model or graph model is required. The most commonly used text vectorization method is TF-IDF. TF-IDF indicates the importance of a word within a document by computing its weight based on relevance [8]. It also suggests that the importance of a term is inversely related to its frequency across multiple documents. The Bag of Word Model (BOW) [9] is a vector representation for unstructured text documents. But BOW does not record the arrangement of words in the sentence nor says anything about how words are associated with sentences. Word embeddings are a technique where individual words are transformed into a

numerical representation of the word (a vector). Word2Vec [10] can make strong estimates about a word's meaning based on its occurrences in the text. The Word2Vec model has the ability to generate real-valued dense vectors for each word. These vectors are capable of capturing linguistic regularities and linear relationships, allowing them to be used in mathematical operations like addition and subtraction. GloVe [11] impact suggests finding the relationship between two words in terms of probability rather than raw counts; Word embedding models use large text corpora, but there are difficulties with rare words that are not common in training corpora [12], Word embedding models have difficulty discriminating between the different meanings of ambiguous words, as most often these models combine different meanings into a single embedding. Moreover, these models analyze words in isolation, ignoring the context in which they occur, which means that two identical words used in different contexts may have the same embedding, which can lead to the loss of semantic information. BERT [13] uses a transformer-based ML model to convert phrases, words, etc. into vectors. The key difference between BERT and TF-IDF is that TF-IDF does not provide the semantic meaning or context of the words whereas BERT does. Also, BERT uses deep neural networks as part of its architecture, meaning that it can be much more computationally expensive than TF-IDF. Matheus A Ferraria *et al*. Investigates how different text representations effect the clustering performance of Artificial Immune Network [14]. The work also investigates BOW, Linguistic Inquiry and Word Count (LIWC) [15], Part of Speech Tagging (PoS Tagging) [16], MRC [17], Doc2Vec [18], Word2Vec [10,19], and SBERT [20].

The graph model is another intermediate representation and a convenient way of representing relationships between entities. The work explains how edge weights are computed by using several similarity measures [21-23]. Due to the inherent ambiguity in natural language, crisp graphs cannot properly represent relationship degrees [24]. In fuzzy graph theory, membership functions play a crucial role in quantifying the degree of relationship between adjacent vertices within the graph. These membership functions typically assign a value between 0 and 1, representing the strength or degree of connection between the vertices. By computing the vertex values based on these membership functions and applying operations such as the min operation between adjacent vertices, the edge values in a fuzzy graph can be determined. This approach helps in modeling uncertain or imprecise relationships within a graph using fuzzy logic concepts [25, 26].

The model being proposed aims to establish a novel document model using fuzzy graphs for efficiently calculating the degree of relationship between two interconnected entities. The focus of the research was on fundamental ideas related to fuzzy sets, operations performed on fuzzy sets, building fuzzy graphs, manipulating fuzzy graphs, sentence ranking techniques, and more. In 1975, Rosenfield introduced fuzzy relations on fuzzy sets and formulated the theory of fuzzy graphs [27]. Zadeh *et al*. discusses the concepts of fuzzy sets, the operations on fuzzy sets, membership function, fuzzification, defuzzification, etc. [28]. Sunitha and Sunil Mathew explain the basics of the connection between nodes in a fuzzy graph, edges as well as cycles, blocks, and cycle connectivity in fuzzy graphs [29]. The paper discusses about fuzzy set theory, fuzzy subsets, fuzzy relation, multi-criteria decision making, etc. [30]. Arya Sebastian *et al*. provide an explanation of the theorems regarding vertex and edge connectivity in fuzzy graphs [31]. Beena G. Kittur talks about calculating eigen values in fuzzy graphs [32]. The discussion by S.Samanta *et.al* covers complete fuzzy graphs, generalized fuzzy graphs, and matrix representation of fuzzy graphs [33]. Cen Zuo *et al*. talked about operations in fuzzy graphs, as well as their applications, etc. [34]. The work shows how fuzzy graph structure can

be used effectively for decision-making [35]. Hypertext Induced Topic Search is a search (HITS) algorithm that ranks web pages based on their relevance and authority [36]. Positional power function [37] and eigen analysis are utilized for determining the ranking of sentences [38]. The work ranks sentences for text summarization using the page rank algorithm [39]. H. S. Wilf has assessed the concept of eigenvectors for ranking to analyze the importance of web pages [40]. The important sentences in the text summarization could also be determined by graph based lexical centrality [41, 42]. The work incorporates improvement in the edge weight scores for the most relevant sentences by considering the relative position of sentences within the document [43]. In this work the communities are constructed in the form of graphs from documents [44]. The text document is converted to graph and summary generation is done by ranking sentences [45]. Abeer Alzuhair *et al.* compared different similarity measures and ranking methods in an undirected weighted graph model [46]. The work builds a correlation graph, in which the vertices are frequent item sets extracted by using association rule mining, and the edge weight represents the correlation between node pairs [47]. The work builds a framework for analyzing long and short-term topics over time and the user's reaction to the topics. The topics were extracted by using the weight calculated for the topic [48].

The sentence is an important linguistic unit of natural language. TF-IDF, BOW, Word2vec, and GloVe are word representations, but the proposed model defined the vectors for each sentence of a document based on special features, as discussed in Sec.3. To enhance the efficiency of the proposed system, we have designed an algorithm for constructing good intermediate document representation in the form of a fuzzy graph, by analyzing the vector representations of each sentence in the input document, with very simple architecture and reasonable computational cost.

Now the problem can be stated as follows:

The objective is to design a fuzzy graph based document model and to prove its effectiveness by applying it to a text processing task such as summarization. For this, significant sentence level features from the input document have to be identified and fuzzy membership functions have to be computed.
The proposed fuzzy graph construction algorithm involves the following steps:

1. Identify the sentence level features for the construction of the fuzzy graph model of the document.
2. Define the fuzzy membership function for each of the identified features.
3. Assign weights for each feature using algorithm 6.
4. Compute the strength of each sentence based on the fuzzy membership function of each identified feature.
5. Construct the fuzzy graph using algorithm 4.
6. Measure the effectiveness of the constructed fuzzy graph using the procedure in Sec.4.

The details follow.

3. The Proposed Solution

The fuzzy graph model construction method is described in this section. To process the text document, the proposed algorithm assigns weights for each sentence based on how well it expresses some of the important information in the document. Significant information in the sentences is called text features.

The proposed system represents the input document D as a vector of sentences S_1, S_2, S_3, S_i, ..., S_n where n is the total number of sentences in D, $1 \leq i \leq n$. The fuzzy graph model of the document is composed of nodes (representing the sentences), and edges representing the relationship between the connected nodes. Each edge weight between the connected nodes of a fuzzy graph G: (σ, μ) represents the strength of the underlying relationship. The weight of a node represents the relative importance of the underlying sentence within the document.

After resolving the coreferences in the input document, the work primarily consists of the following steps:

i Based on the requirement of the application, identify the text features
ii Assign weights to each feature based on feature priorities
iii Construct the matrix: sentence X feature-weight by analyzing each sentence of the input document
iv Compute the matrix: sentence X feature-strength
v Construct the fuzzy graph, G from the sentence X feature-strength matrix
vi Evaluate the effectiveness of the constructed fuzzy graph

3.1 Coreference Resolution

Two noun phrases are said to be coreferring to each other, if both of them resolve unique referent unambiguously. It is a two-step process namely, antecedent identification and corresponding mention identification. Resolving coreferent affects the results of NLP application to a great degree and can be applied to a variety of NLP tasks such as text understanding, information extraction, machine translation, sentiment analysis, document summarization, etc. Humans use many kinds of knowledge and signals to resolve ambiguity. We need to know a lot of context and information to figure out what a word actually refers to.

The proposed model adopts the work, which is a pipeline extension for spaCy 2.1+ that resolves coreference clusters using a neural network. The model uses word embedding of several words inside and around each mention to obtain the feature representation for each mention. Subsequently, these feature values are loaded into two neural networks. The first of these networks gives us a score for each pair of mentions, a possible antecedent, and the second network computes a score for each mention having no antecedent. All computed scores are compared to get the highest score which decides the correct mention of an antecedent. It is observed that there is a significant improvement in the performance of the proposed system after performing coreference resolution [49].

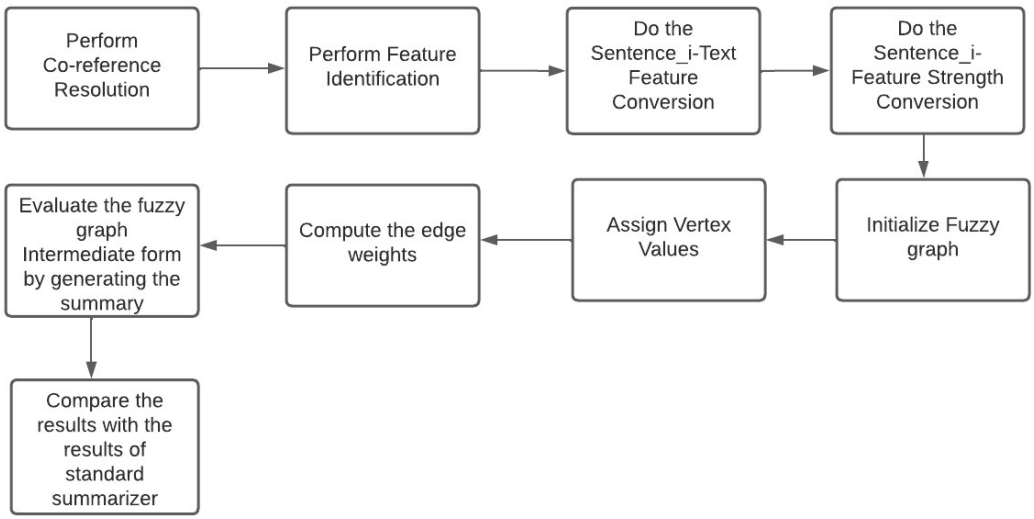

Fig. 1. System Design

3.2 Feature Extraction and Membership function definition

The text features required for the specific application were extracted from each sentence of the input text to compute the strength of each vertex of the fuzzy graph model, so as to extract and contribute relevant information for the application. The following significant features were identified based on the hypothesis [50] and the membership function was defined for each text feature by the intuition method [51].

The following membership functions were defined for the construction of the proposed model.

1. Sentence position:

This hypothesis states that certain positions within the sentence such as the beginning, middle, or last of the document may carry information required for the construction of the fuzzy graph model of that document.

$$\mu\left(f_{sent\_position}\right) = \begin{cases} 1 \; if \; i = 1, \frac{n}{2} \; or \; n, \\ where \; n \; is \; the \; number \; of \\ sentences \; in \; the \\ \; input \; document. \\ \frac{1}{i} \; otherwise \end{cases} \tag{1}$$

where $1 \leq i \leq n$; n= No: of sentences in the input document.

2. Sentence length:

In most cases, sentences with fewer words contain less information. The model assumes that longer sentences carry more relevant information.

$$\mu\left(f_{sent\_length}\right) = \frac{\left|S_i\right|}{\left|S_{longest}\right|} \tag{2}$$

where $|S_i|$ is the length of i^{th} sentence and $|S_{longest}|$ is the length of longest sentence among all the sentences.

3. Numerical data:

It is observed that sentences that contain numerical entities like date, year, amount, etc., carry relevant and critical information. The work assumes that this hypothesis strengthens fuzzy graph vertex.

$$\mu\left(f_{num\_data}\right) = \frac{N_{num\_data}\left(S_i\right)}{\left|S_i\right|} \tag{3}$$

where $N_{num\_data}(S_i)$ is the total number of numerical information in the sentence (S_i) and $|S_i|$ is the length of i^{th} sentence in words.

4. Thematic Word:

A few words occur frequently; most probably, they represent the context of the document, and sentences containing these words are considered important. The model expects that these sentences will make a significant contribution to the construction of the fuzzy graph.

$$\mu\left(f_{Them\_word}\right) = \frac{N_{Thematic}\left(S_i\right)}{\left|S_i\right|} \tag{4}$$

where $N_{Thematic}(S_i)$ is the number of thematic words in a sentence S_i and $|S_i|$ is the length of the i^{th} sentence in words.

5. Proper noun:

The sentence containing the proper noun, named entity, carries information about a person, place, or thing. Therefore, these sentences may play a major role in the construction of the fuzzy graph.

$$\mu\left(f_{pro\_noun}\right) = \frac{N_{pro\_noun}\left(S_i\right)}{\left|S_i\right|} \tag{5}$$

where $N_{pro\_noun}(S_i)$ is the number of proper nouns in S_i and $|S_i|$ is the length of the sentence S_i in words.

6. Title feature:

A sentence containing words in the title must be considered as important. The membership function for the sentence S_i is calculated as the ratio of the number of title words occurring in the sentence S_i to the total number of words in the title T, where $1 \leq i \leq n$. It is defined as:

$$\mu\left(f_{tit\_feature}\right) = \frac{\left|S \cap T\right|}{\left|T\right|} \tag{6}$$

7. Upper case:

In this work, the sentences carrying uppercase words have more priority than the other sentences.

$$\mu\left(f_{upp\_case}\right) = \frac{N_{Upp}\left(S_i\right)}{lar\left(N_{Upp}(S)\right)} \tag{7}$$

where $N_{Upp}(S_i)$ is the total count of uppercase words in S_i and $lar(N_{Upp}(S))$ denotes the largest count of uppercase words among the sentences S_i in the document, where $1 \leq i \leq n$.

8. Cue phrases:

Certain words and phrases like ‘significant’, ‘in this paper’, ’we show’, etc., explicitly signal importance, and the sentences containing these words have to be extracted to strengthen the vertex weights and edge weights in the graph.

$$\mu\left(f_{cue\_phrases}\right) = \begin{cases} 1 \text{ if cue word is present} \\ 0 \text{ if cue word is not present} \end{cases} \tag{8}$$

9. Sentence-sentence similarity:

This feature is used to identify the central aspects of the document.

$$\mu\left(f_{sent\_similarity}\right) = \frac{\sum Sim\left(S_i, S_j\right)}{max\left(Sim\left(S_i, S_j\right)\right)} \tag{9}$$

where max (Sim (S_i, S_j)) is the maximum similarity between i^{th} sentence and j^{th} sentence.

3.3 Feature Weights

The 9 features used by the model are divided into two sets, namely *feature_set*1 and *feature _set*2. The priority of elements in *feature_set*1 is greater than those in *feature _ set2*. The members of *feature_set*1 are $f_{cue\_phrases}$, f_{num_data}, f_{upp_case}. The specialty of these features is that the features rarely occur in sentences, but the infrequent features provide a high impact on the applications. As in TF-IDF the importance of a feature is inversely related to its frequency across document. The priority of the features in *feature_set*1 was assigned by analyzing the sentences in the text document and their summary. The weights for features in *feature_set*1 were assigned in the order of priority are f_{num_data}, f_{upp_case}, $f_{cue\_phrases}$. The weights assigned to each feature are in between 0 and 1. The value of w_i associated with each feature, $feature_i$ reflects the importance of the feature over another. The remaining 6 features are included in the *feature_set*2. The feature priorities of features in *feature_set*2 are determined by Algorithm 6.

3.4 Feature Strength Computation for each Sentence

The strength of each sentence is calculated as a weighted sum of feature values. The feature strength calculation of each sentence in the text document is shown in the Algorithm 1.

Algorithm 1 Sentence Feature Strength Computation

Input: Text document.
Output: Feature strength of each sentence of the text document.

```
        /*n= No: of sentences in the text document. */
      /* k be the no: of text features identified by the model. */
1         k=9
2         i=1
3         while i≤ n do
4            sentence_feature_strength=0
5                        j=1
6                    while j≤ k do
      /* w(fj) is the weight assigned to the j^th feature. */
7                 w(fj)=feature_weight(fj)
     /* value(fj) is the feature value of j^th feature computed by
          the membership function*/
8                        sentence_feature_strength= sentence_feature_strength+
                          (value(fj)*w(fj))
9                       j=j+1
10                   end while
11    sentence _feature _strength[i] = sentence_feature_strength
12                i=i+1
13        end while
```

3.4.1 Fuzzy graph vertex initialization

For the construction of the proposed fuzzy graph model, the vertices of the fuzzy graph have to be initialized. The initialization of fuzzy graph vertices is shown in Algorithm 2.

Algorithm 2 Fuzzy-graph-vertex-initialization

Input: n= Number of sentences in the input document.
Output: Fuzzy Graph G initialized with 'n' vertices, where n is the number of sentences in the input document, and vertex values initialized to 0.

```
   /* n=No: sentences in the input document. */
1        for i= 1 to n do
2            for each sentence (Si)
3                Create vertex (Vi)
4                    Vertex_value (Vi)=0
5            end for
6        end for
```

3.4.2 Vertex value assignment

Each sentence in the text document was analyzed for building the fuzzy graph. For each sentence, feature identification and its membership functions were defined. The membership value lies between [0, 1]. The membership values of each vertex were initialized as proposed by [35]. Edge membership values between two vertices are then taken based on vertex membership values as proposed by [31]. In this study, sentences were considered as the vertices of the fuzzy graph. The vertex value of the fuzzy graph was determined using the sentence_feature_strength value. The membership value of each sentence (S_i) was calculated using the Algorithm 3 and it is assigned to the i^{th} vertex of the fuzzy graph.

Algorithm 3 Fuzzy graph vertex value assignment

Input: Fuzzy graph G with n vertices
Output: Vertex value assignment of Fuzzy graph G

```
   /*n= No: of sentences in the input document*/
1  max_strength=max (sentence_feature_strength(S1,S2…,Sn))
2   for i= 1 to n
3     V(i)=sentence_feature_strength[i]/max_strength
```

where S_i is the i^{th} sentence and s*entence_feature_strength* (S_i) is the feature strength of the sentence S_i

```
4  end for
```

3.5 Fuzzy Graph Construction

For the construction of the proposed fuzzy graph model, the vertices of the fuzzy graph have to be initialized. The initialization of fuzzy graph vertices is shown in Algorithm: 2. The membership value of each sentence (S_i) is computed using Algorithm 3. The membership values of the vertex were initialized first as proposed by [35]. The edge membership value

between two vertices of a fuzzy graph were computed based on vertex membership values as proposed by [31]. The fuzzy graph is represented by the following function G: (σ, μ) where σ is a fuzzy subset of X and μ is a symmetric relation on σ i.e. $\sigma: X \rightarrow [0,1]$, μ X x X $\rightarrow [0,1]$, such that $\mu(x,y) \leq \sigma(x) \wedge \sigma(y)$ $\forall x, \forall y$ in X. The membership strength of the sentence S_i is assigned as the vertex values v_i $\forall v_i$. The edge weights between the vertices (v_i, v_j) where $1 \leq i \leq n$, $1 \leq j \leq n$, $i \neq j$ of the fuzzy graph are computed using the fuzzy intersection operation. When two fuzzy sets overlap, they indicate how closely they are related. The lower membership value in both sets of each element is assigned as the degree of membership. Let A and B be two fuzzy sets, The intersection operation and the value of the membership function is expressed as: $\mu_{(A, B)}(x) = \min \{\mu_A(x), \mu_B(x)\}$ [51]. **Fig. 2** shows the fuzzy graph $G = (\sigma, \mu)$ constructed using the Algorithm 4 for the document having 5 sentences. $\mu(v_i, v_j) \leq \sigma(v_i) \wedge \sigma(v_j)$ $\forall v_i, \forall v_j \in V$, $i, j = 1, 2, 3, ...n$, where n is the number of vertices in G.

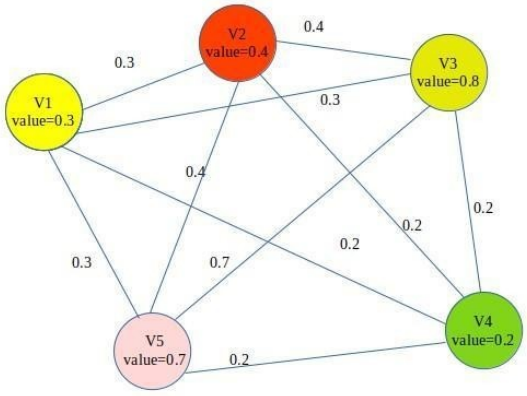

Fig. 2. Fuzzy Graph

Algorithm 4 Fuzzy graph Computation

Input: n number of vertices with vertex value initialized by Algorithm 3.
Output: The fuzzy graph G :(σ, μ)

```
    /* Initialize the number of vertices as the sentence count of the input document. */
    /* n = No: of the sentence in the text document*/
1       for i = 1 to n -1
2           for j = i+1 to n
  /*  ev(i, j) is the edge weight between vertices vi and vj, and it is obtained after
    applying fuzzy intersection operation */
3               for each edge e, between <V(i), V(j)> do
4                   ev(i,j) = min(μv(i)(x), μv(j) (x))
5               end for
6           end for
7       end for
```

3.6 Weighted Feature Inter-Sentence Correlation

Any graph can be represented in the form of a matrix [52]. The Weighted Feature Inter-Sentence matrix S computed from the fuzzy graph G is a symmetric one. In the adjacency matrix, the rows and columns are represented by the sentences of the document. Each entry in S_{ij} represents the degree of similarity between sentences S_i and S_j and its values are in the range [0,1]. The weighted-feature inter-sentence similarity was captured using the inter-sentence correlation matrix.

3.7 Assessing the effectiveness of the model

The effectiveness of the constructed fuzzy graph was evaluated by inputting it into a text summarizer that uses eigen analysis. The summary thus obtained was compared with the summaries generated by three standard summarizers and summaries from Kaggle Dataset.

Algorithm 5 Summary Generation

Input: Weighted-Feature, Inter-Sentence Correlation matrix (WFISC) constructed from fuzzy graph G
Output: Document summary, S_{imp} for the input document D.
1 Read the WFISC matrix.
2 Compute the eigen values of WFISC *matrix*
3 Compute the eigen vectors of *WFISC*
4 Return the sentence S_{imp}, corresponding to the largest eigen vector.
5 By considering the sentences S_{imp}, from the document D, output the summary.

3.8 Priority Computation for Features

The relative weights need to be assigned for each feature based on the feature priorities so as to improve the quality of the application. The feature priorities were identified using the algorithm: 6. The feature priorities of the elements in the *feature_set*2 in the decreasing order are fsent_length , fsent_position , ftit_feature , fpro_noun , fthem_word , fsent_similarity.

4. Experimental Results and Discussion.

4.1 Dataset

The dataset is made up of samples from news articles. The Kaggle dataset [54] for news articles is used for the model evaluation. The generated summary was N% of the input document, where N is the percentage of summary required by the user. The 100 documents and its summary were taken from the Kaggle dataset. The proposed system and QuillBot's online summary tool were both used to generate the summary from the same dataset. The 30 input documents and the summary generated by the three systems were distributed to 25 unbiased humans and evaluated by them.

Algorithm 6 Feature priority Computation and weight assignment for feature

Input: The 6 features in the feature_set2.
Output: Assignment of feature priorities.

```
    /* m is the No: of input documents in the dataset*/
1    for d = 1 to m do
    /* n = No. of sentences in the text document*/
2      for j= 1 to n do
    /* f2n is the No: features in the feature_set2*/
3          for i = 1 to f2n do
4              Sent_feature_strength[j,i]=membership value(feature_i)
5          end for
6      end for
7   Initialize the fuzzy graph vertex using the algorithm
      Fuzzy-graph-vertex-initialization for the text document-d
8   Assign the value to each vertex of the fuzzy graph by the algorithm
     Fuzzy graph-vertex value-Assignment
9   Construct the fuzzy graph by using the Fuzzy graph computation algorithm.
10  Generate the summary of the document-d by Algorithm 5
11  Evaluate Rouge-1, Rouge-2, Rouge-L values.
12  Extract the precision, recall, and F Measure values
13  end for
14  for i=1 to m do
15    precision=precision+ (precision value(i^th document))
16    recall=recall+ (recall value(i^th document))
17    F-measure=F-measure+ (F-measure value(i^th document))
18  end for
19     avg-precision=precision/m
20     avg-recall=recall/m
21     avg-F-measure value=F-measure/m
22     Sort the average precision values
23    Assign weights to the features based on sorted average precision
      values with the highest weight assign to the feature having the
      highest average precision value.
```

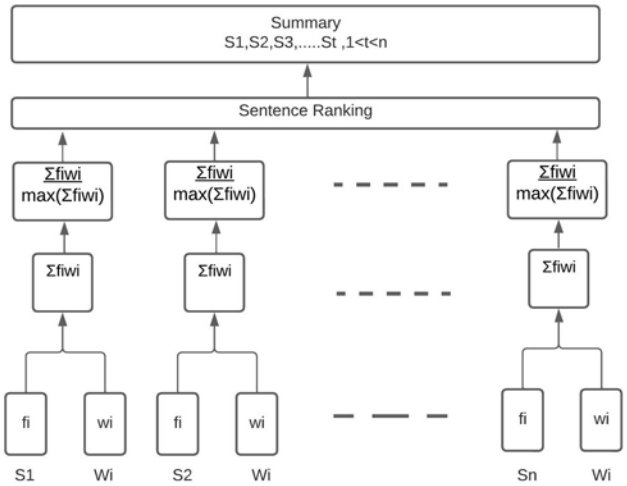

Fig. 3. Fuzzy graph Summarizer architecture

4.2 Evaluation of the Model

We used an eigen-analysis-based text summarizer for evaluating the effectiveness of our model. The model was evaluated by unbiased human judges using different quality metrics. The model was also evaluated by ROUGE (Recall-Oriented Understudy for Gisting Evaluation) [53]. ROUGE evaluated the summary generated by the proposed system with the summaries generated by the state-of-the-art summarizers such as Similarity graph Summarizer, EMM summarizer, BERT Extractive Summarizer and summaries from Kaggle Dataset. The reference summary selected for the evaluation of the model was the one by QuillBot online summarizer. It is an online summarizing tool for documents like news articles, research papers, etc. Kaggle dataset contains 4515 samples and their summaries. The summary generated by the proposed model was evaluated using ROUGE-1, ROUGE-2, and ROUGE-L metrics also.

4.3 Evaluation of the Model using One-Way ANOVA Test

In order to assess the results of the Fuzzy graph Summarizer, the One-Way ANOVA statistical test was performed. It tells us if there is any significant difference between the summary mean of the three groups. In this work, the summaries generated by using three different algorithms were subjected to comparison. The One-way ANOVA test was performed on a summary generated by EMM summarizer [55], QuillBot online summary, and summary generated by the proposed Fuzzy graph Summarizer. The null hypothesis H_0 states that after evaluating the three different summary groups by unbiased humans, there is no difference between the means of these summary groups. The alternate hypothesis H_a states that after evaluating the three summary groups by unbiased humans, there is a significant difference between the means of summary groups and also there is a significant improvement in the mean value of the proposed model compared to similarity graph summary and summary generated by EMM summarizer. The One-Way ANOVA test was performed over a randomly selected population of 30 samples of news articles and their summaries. The calculated P-value was less than 0.05. Results after the One-way ANOVA-test proved that the alternate hypothesis is strongly true.

4.4 Comparison with Other Summarization Models

The work compares the F-measure and Recall of graph based summarizers that use different sentence ranking methods for extractive summarization [56]. This comparison shows that the summarizer that uses Page Rank algorithm for sentence ranking reports the highest accuracy [56]. So the summary generated by the Fuzzy graph Summarizer was compared with the one generated by the Similarity graph Summarizer which uses cosine similarity and Page Rank. The Fuzzy graph Summarizer's performance, as measured by F1 scores for ROUGE, is: Rouge-1 - 0.72, Rouge-2 - 0.66, Rouge-L 0.71, compared to the Similarity graph Summarizer's F1 scores of 0.5, 0.4, and 0.53 for Rouge-1, Rouge-2, and Rouge-L respectively. The comparison showed that the performance of proposed Fuzzy graph Summarizer to be superior to that of the Similarity graph Summarizers, as shown in **Fig. 4**.

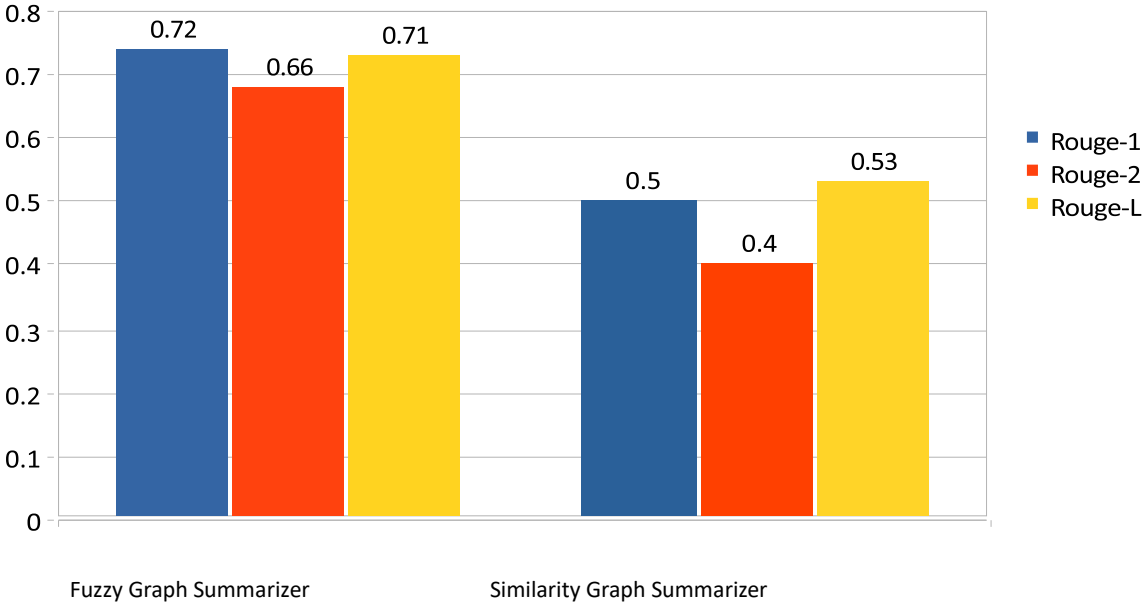

Fig. 4. F-measure based comparison of the proposed Fuzzy graph Summarizer and Similarity graph Summarizer

In order to assess the performance of the Fuzzy graph Summarizer, the results were also compared with those from other text summarizers, including the EMM summarizer, summary from the Kaggle dataset. The reference summary was taken as the summary generated by the QuillBot online summarizer. The system summaries for comparison were obtained from Fuzzy graph Summarizer, summary from Kaggle dataset and by the EMM Summarizer. Rouge-1 refers to the overlap of unigrams between model summary and reference summary. Rouge-2 refers to the overlap of bigrams between model summary and reference summary. Rouge-L measures longest matching sequence of words using longest common subsequence. **Fig. 5** and **Fig. 6** provide information about the average precision and recall value of different summarization models obtained after evaluating the summaries using Rouge-1, Rouge-2 and Rouge-L and it was observed that the Fuzzy graph Summarizer which utilizes the proposed fuzzy graph, yields good performance compared to other models.

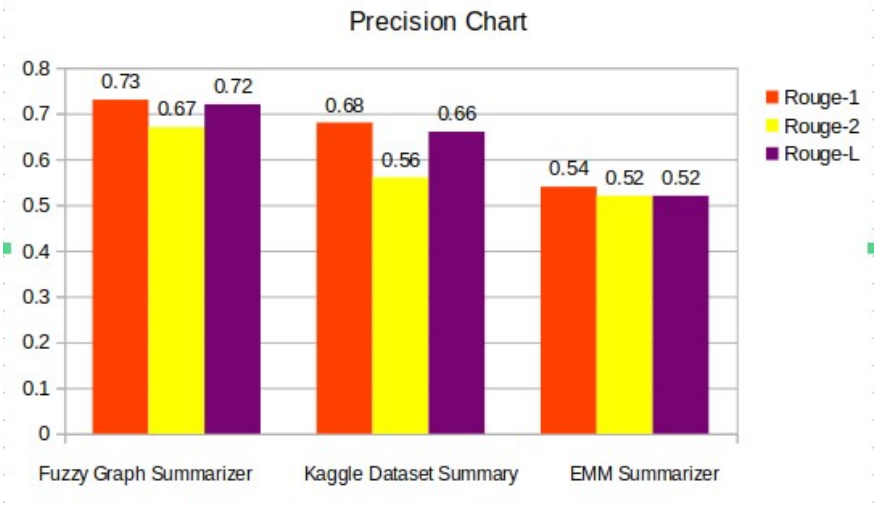

Fig. 5. Comparison of precision values for different summarization models.

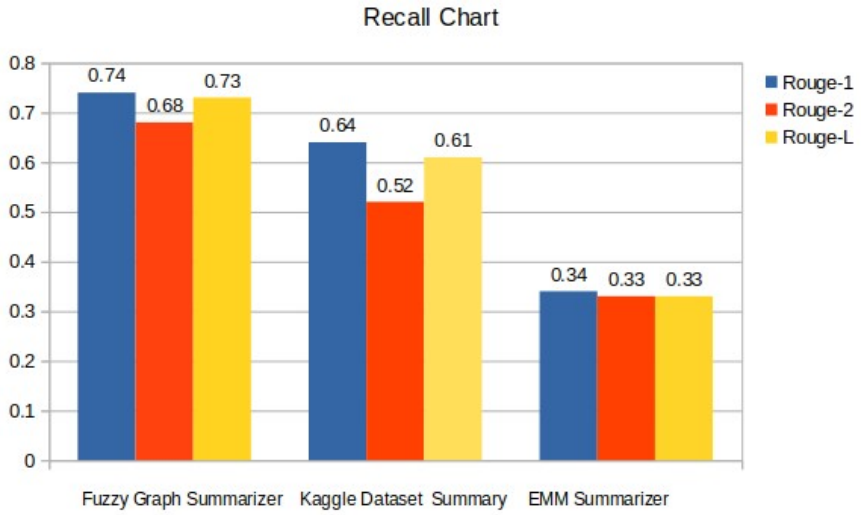

Fig. 6. Comparison of recall values for different summarization models.

The evaluation of the 100 documents from the Kaggle dataset involved a thorough analysis using Fuzzy graph Summarizer and the BERT Extractive Summarizer [57]. **Fig. 7** illustrates the comparison of the F-measure scores between two summarization techniques. The evaluation showed that the summary generated by the proposed Fuzzy graph Summarizer is better than the latter in the sense that the generated summary was informative, understandable, and also preserving the chronological order.

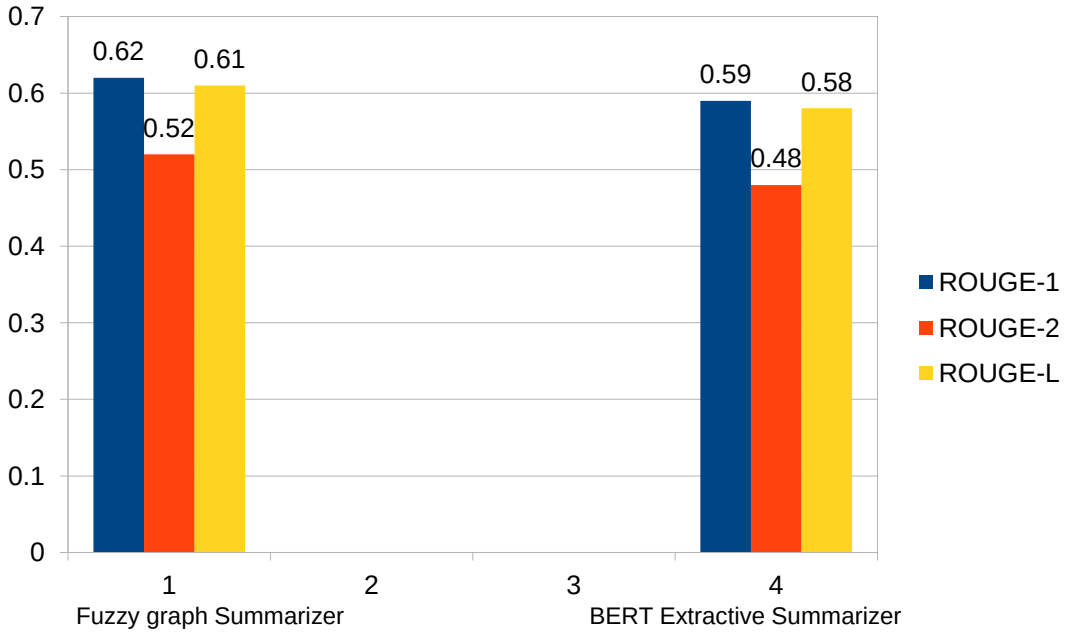

Fig. 7. F- Measure comparison of Fuzzy graph Summarizer and BERT Extractive summarizer

5. Limitations and Suggestions for Future Work

The proposed model considered only nine significant sentence features. Incorporating significant sentence-level features such as *date* and *quotation marks* will enhance the effectiveness of the model. Employing the membership function to compute semantic similarity between sentences can significantly enhance the proposed system's ability to model text.

The concepts used for constructing the fuzzy graph from text can be used for a variety of NLP applications, such as quantifying the relationship between entities, identifying important paragraphs in a text document, question-answering, opinion mining, relation extraction, etc,. The effectiveness of the proposed model can be explored in such NL tasks as a future work.

6. Conclusion

The paper proposes a novel algorithm for fuzzy graph based document modeling and shows that the model is an effective one by applying it to the text summarization task. For this, nine sentence features are identified from the input document. Appropriate fuzzy membership functions are defined for each feature. Each vertex in the graph corresponds to a sentence in the document. Each edge has a weight computed by considering the membership values of the nine features identified. The usefulness of the constructed fuzzy graph as an intermediate representation of the document is assessed by applying it to the Fuzzy graph Summarizer, which uses eigen analysis for ranking the sentences to generate a meaningful extractive summary. The performance of the proposed model was evaluated by comparing the quality of the summary thus generated, with the summaries generated by the state-of-the-art online summarizer (QuillBot) and the Kaggle dataset summary. Measures such as ROUGE-1,

ROUGE-2, and ROUGE-L are used for evaluation. The summarizer that uses fuzzy graph as intermediate representation showed better performance than the other summarizers, which proves that the proposed fuzzy graph model of text documents is very effective.

References

[1] Shahzad Qaiser and Ramsha Ali, "Text Mining: Use of TF-IDF to Examine the Relevance of Words to Documents," *International Journal of Computer Applications*, vol.181, no.1, pp.25-29, 2018. Article (CrossRef Link)

[2] Yin Zhang, Rong Jin, and Zhi-Hua Zhou, "Understanding bag-of-words model: a statistical framework," *International journal of machine learning and cybernetics*, vol.1, pp.43-52, 2010. Article(CrossRef Link)

[3] Tomas Mikolov, Kai Chen, Greg Corrado, Jeffrey Dean, "Efficient Estimation of Word Representations in Vector Space," in *Proc. of International Conference on Learning Representations (ICLR)*, 2013. Article(CrossRef Link)

[4] Jeffrey Pennington, Richard Socher, and Christopher Manning, "Glove: Global vectors for word representation," in *Proc. of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pp.1532-1543, 2014. Article (CrossRef Link)

[5] Jacob Devlin et al., "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," in *Proc. of 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, vol.1, pp.4171-4186, 2019. Article(CrossRef Link)

[6] Roi Blanco and Christina Lioma, "Graph-based term weighting for information retrieval," *Information Retrieval*, vol.15, pp.54-92, 2012. Article(CrossRef Link)

[7] Nurfarhana Hassan and Tahir Ahmad, "A review on taxonomy of fuzzy graph," *Malaysian Journal of Fundamental and Applied Sciences*, vol.13, no.1, pp.6-13, 2017. Article(CrossRef Link)

[8] Denis Eka Cahyani and Irene Patasik, "Performance comparison of TF-IDF and Word2Vec models for emotion text classification," *Bulletin of Electrical Engineering and Informatics (BEEI)*, vol.10, no.5, pp.2780-2788, 2021. Article(CrossRef Link)

[9] Nisha V M and Ashok Kumar R, "Implementation on Text Classification Using Bag of Words Model," in *Proc. of the second international conference on emerging trends in science & technologies for engineering systems (ICETSE-2019)*, 2019. Article(CrossRefLink)

[10] Derry Jatnika, Moch Arif Bijaksana, and Arie Ardiyanti Suryani, "Word2Vec Model Analysis for Semantic Similarities in English Words," *Procedia Computer Science*, vol.157, pp.160-167, 2019. Article(CrossRef Link)

[11] Deepak Suresh Asudani, Naresh Kumar Nagwani and Pradeep Singh, "Impact of word embedding models on text analytics in deep learning environment: a review," *Artificial Intelligence Review*, vol.56, pp.10345-10425, 2023. Article(CrossRef Link)

[12] Derry Jatnika, Moch Arif Bijaksana, and Arie Ardiyanti Suryani, "Word2Vec Model Analysis for Semantic Similarities in English Words," *Procedia Computer Science*, vol.157, pp.160-167, 2019. Article(CrossRef Link)

[13] Jacob Devlin, Ming-Wei Chang, Kenton Lee, Kristina Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," in *Proc. of North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT 2019)*, pp.4171-4186, 2019. Article(CrossRef Link)

[14] Matheus A. Ferraria, Vinicius A. Ferraria, and Leandro N. de Castro, "An Investigation Into Different Text Representations to Train an Artificial Immune Network for Clustering Texts," *International Journal of Interactive Multimedia and Artificial Intelligence*, vol.8, no.3, pp.55-63, 2023. Article(CrossRef Link)

[15] Yla R. Tausczik and James W. Pennebaker, "The Psychological Meaning of Words: LIWC and Computerized Text Analysis Methods," *Journal of Language and Social Psychology*, vol.29, no.1, pp.24-54, 2010. Article(CrossRef link)

[16] Kristina Toutanova, Dan Klein, Christopher D. Manning, and Yoram Singer, "Feature-rich part-of-speech tagging with a cyclic dependency network," in *Proc. of NAACL '03: Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology*, vol.1, pp.173-180, 2003. Article (CrossRef Link)

[17] Michael Wilson, "MRC psycholinguistic database: Machine-usable dictionary, version 2.00," *Behavior Research Methods, Instruments, & Computers*, vol.20, pp.6-10, 1988. Article(CrossRef Link)

[18] Quoc V. Le and Tomas Mikolov, "Distributed Representations of Sentences and Documents," in *Proc. of ICML'14: Proceedings of the 31st International Conference on International Conference on Machine Learning*, vol.32, no.2, pp.1188-1196, 2014. Article(CrossRef Link)

[19] Derry Jatnika, Moch Arif Bijaksana, and Arie Ardiyanti Suryani, "Word2Vec Model Analysis for Semantic Similarities in English Words," *Procedia Computer Science*, vol.157, pp.160-167, 2019. Article(CrossRef Link)

[20] Anna Rogers, Olga Kovaleva, and Anna Rumshisky, "A Primer in BERTology: What We Know About How BERT Works," *Transactions of the Association for Computational Linguistics*, vol.8, no.5, pp.842-866, 2020. Article(CrossRef Link)

[21] Reinert Yosua Rumagit, Nina Setiyawati, and Dwi Hosanna Bangkalang, "Comparison of Graph-based and Term Weighting Method for Automatic Summarization of Online News," *Procedia Computer Science*, vol.157, pp.663-672, 2019. Article(CrossRef Link)

[22] Ahmad Rawashdeh and Anca L. Ralescu, "Similarity Measure for Social Networks – A Brief Survey," in *Proc. of Modern AI and Cognitive Science Conference (MAICS)*, vol.1353, pp.153-159, 2015. Article(CrossRef Link)

[23] Çağatay Tülü, Umut Orhan, and Erhan Turan, "Semantic Relation's Weight Determination on a Graph Based WordNet," *Gümüşhane University Journal of Science and Technology*, pp.67-78, 2018. Article(CrossRef Link)

[24] Ahmed Hamza Osman and Omar Mohammed Barukub, "Graph-Based Text Representation and Matching: A Review of the State of the Art and Future Challenges," *IEEE Access*, vol.8, pp.87562-87583, 2020. Article(CrossRefLink)

[25] Xiaolong Shi et al., "A Novel Domination in Vague influence Graphs with an Application," *Axioms*, vol.13, no.3, 2024. Article(CrossRef Link)

[26] Muzzamal Sitara, Muhammad Akram, and Muhammad Yousaf Bhatti, "Fuzzy Graph Structures with Application," *Mathematics*, vol.7, no.1, 2019. Article(CrossRef Link)

[27] Azriel Rosenfeld, "Fuzzy graphs," *Fuzzy Sets and their Applications to Cognitive and Decision Processes*, pp. 75-95, 1975. Article(CrossRef Link)

[28] L.A. Zadeh, "Fuzzy Sets and Their Application to Pattern Classification and Clustering Analysis," in *Proc. of Classification and clustering: Proceedings of an Advanced Seminar Conducted by the Mathematics Research Center*, pp.251-299, 1977. Article(CrossRef Link)

[29] M.S Sunitha and Sunil Mathew, "Fuzzy Graph Theory: A Survey," *Annals of Pure and Applied Mathematics*, vol.4, no.1, pp.92-110, 2013. Article(CrossRef Link)

[30] H.-J. Zimmermann, Fuzzy set theory—and its applications, *Springer Science & Business Media*, 2011. Article(CrossRef Link)

[31] Arya Sebastian, John N Mordeson, and Sunil Mathew, "Generalized Fuzzy Graph Connectivity Parameters with Application to Human Trafficking," *Mathematics*, vol.8, no.3, 2020. Article(CrossRef Link)
[32] Beena G. Kittur, "Eigen Values of Complete Fuzzy Graphs," *International Journal of Fuzzy Mathematics and Systems*, vol.2, no.3, pp.293-296, 2012. Article(CrossRef Link)
[33] Sovan Samanta, Biswajit Sarkar, Dongmin Shin, Madhumangal Pal, "Completeness and regularity of generalized fuzzy graphs," *SpringerPlus*, vol.5, no.1, 2016. Article(CrossRef Link)
[34] Cen Zuo, Anita Pal, and Arindam Dey, "New Concepts of Picture Fuzzy Graphs with Application," *Mathematics*, vol.7, no.5, 2019. Article(CrossRef Link)
[35] Ali N. A. Koam, Muhammad Akram, and Peide Liu, "Decision-Making Analysis Based on Fuzzy Graph Structures," *Mathematical Problems in Engineering*, vol.2020, 2020. Article(CrossRef Link)
[36] Khushboo S. Thakkar, R.V. Dharaskar, and M.B. Chandak, "Graph-Based Algorithms for Text Summarization," in *Proc. of 2010 3rd International Conference on Emerging Trends in Engineering and Technology*, pp.516-519, 2010. Article(CrossRef Link)
[37] P. Jean-Jacques Herings, Gerard van der Laan, and Dolf Talman, "The positional power of nodes in digraphs," *Social Choice and Welfare*, vol.24, pp.439-454, 2005. Article(CrossRef Link)
[38] O.K. Reshma and P.C. Reghu Raj, "Paragraph Ranking Based on Eigen Analysis," *Procedia Computer Science*, vol.46, pp.532-539, 2015. Article(CrossRef Link)
[39] Rada Mihalcea and Paul Tarau, "TextRank: Bringing Order into Text," in *Proc. of the 2004 Conference on Empirical Methods in Natural Language Processing*, pp.404-411, 2004. Article(CrossRef Link)
[40] Herbert S. Wilf, "Searching the web with eigenvectors," *COMAP UMAP Journal Article*, 2001. Article(CrossRef Link)
[41] Linton C. Freeman, "Centrality in social networks conceptual clarification," *Social Networks*, vol.1, no.3, pp.215-239, 1978. Article(CrossRef Link)
[42] Güneş Erkan and Dragomir R. Radev, "LexRank: Graph-based Lexical Centrality as Salience in Text Summarization," *Journal of Artificial Intelligence Research*, vol.22, pp.457-479, 2004. Article(CrossRef Link)
[43] Xiaojun Wan, "An exploration of document impact on graph-based multi-document summarization," in *Proc. of EMNLP '08: Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pp.755-762, 2008. Article(CrossRef Link)
[44] Linhong Zhu et al., "Graph-based informative-sentence selection for opinion summarization," in *Proc. of the ASONAM '13: Proceedings of the 2013 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*, pp.408-412, 2013. Article(CrossRef Link)
[45] Saif alZahir, Qandeel Fatima and Martin Cenek, "New graph-based text summarization method," *2015 IEEE Pacific Rim Conference on Communications, Computers and Signal Processing (PACRIM)*, pp.396-401, 2015. Article(CrossRef Link)
[46] Abeer Alzuhair and Mohammed Al-Dhelaan, "An Approach for Combining Multiple Weighting Schemes and Ranking Methods in Graph-Based Multi-Document Summarization," *IEEE Access*, vol.7, pp.120375-120386, 2019. Article(CrossRef Link)
[47] Elena Baralis et al., "GraphSum: Discovering correlations among multiple terms for graph-based summarization," *Information Sciences*, vol.249, pp.96-109, 2013. Article(CrossRef Link)
[48] Danilo Cavaliere et al., "Emotion-Aware Monitoring of Users' Reaction With a Multi-Perspective Analysis of Long- and Short-Term Topics on Twitter," *International Journal of Interactive Multimedia and Artificial Intelligence*, vol.8, no.4, pp.166-175, *2023*. Article(CrossRef Link)
[49] Thomas Wolf, State-of-the-art neural coreference resolution for chatbots, 2017. Article(CrossRef Link)
[50] S.A. Babar and Pallavi D. Patil, "Improving Performance of Text Summarization," *Procedia Computer Science*, vol.46, pp.354-363, 2015. Article(CrossRef Link)

[51] S. N. Sivanandam and S. N. Deepa, Principles of soft computing (with CD), John Wiley & Sons, 2007. Article(CrossRef Link)
[52] John Clark and Derek Allan Holton, A first look at graph theory. Allied Publishers, 1995. Article(CrossRef Link)
[53] Chin-Yew Lin, "ROUGE: A Package for Automatic Evaluation of Summaries," *Text summarization branches out*, pp.74-81, 2004. Article(CrossRef Link)
[54] https://www.kaggle.com/datasets/sunnysai12345/news-summary.
[55] M R Aswathy and P.C Reghu Raj, "Eigen Analysis Based Document Summarization," in *Proc. of 2018 International CET Conference on Control, Communication, and Computing (IC4)*, pp.399-403, 2018. Article(CrossRef Link)
[56] Kastriot Kadriu and Milenko Obradovic, "Extractive approach for text summarisation using graphs," *arXiv:2106.10955*, 2021. Article(CrossRef Link)
[57] Shehab Abdel-Salam and Ahmed Rafea, "Performance Study on Extractive Text Summarization Using BERT Models," *Information*, vol.13, no.2, 2022. Article(CrossRef Link)

Aswathy M R is a Research Scholar at Government Engineering College, Palakkad, India. Completed M.Tech from Vidya Academy of Science and Technology under Calicut University, Kerala, India. She is currently engaged in the research on fuzzy graph modeling of text documents.
E-mail: aswathy.moozhiyil@gmail.com

Dr. P.C Reghu Raj, Principal, Govt. Engineering College, Kozhikode, Kerala, India under APJKTU. Completed M.Tech from CUSAT (1993) Cochin and Ph.D. from Indian Institute of Technology Madras (2004).
E-mail: pcreghu@gmail.com

Dr. Ajeesh Ramanujan is an Associate professor in the Department of Computer Science at the College of Engineering, Trivandrum, India. Completed Ph.D. from Indian Institute of Technology, Madras, Chennai,India.
Email: ajeeshramanujan@gmail.com