Designing of Fuzzy Inference System to improve text summarization based on fuzzy logic

Gagandeep Singh
Department of Computer Science and Engineering
ABV-IIITM Gwalior
Gwalior, India
gagandeepamolsandhu9878@gmail.com

Bhavana Mekala
Department of Computer Science and Engineering
ABV-IIITM Gwalior
Gwalior, India
bhavanamekala1029@gmail.com

Nelluri Pavithra Sai Lakshmi
Department of Computer Science and Engineering
ABV-IIITM Gwalior
Gwalior, India
pavithrasainelluri1@gmail.com

Santosh Singh Rathore
Department of Computer Science and Engineering
ABV-IIITM Gwalior
Gwalior, India
santoshs@iiitm.ac.in

***Abstract*—Automatic text summarization (ATS) is becoming increasingly important due to the ever-increasing amount of textual content available online. ATS techniques can be broadly categorized as extractive or abstractive. While many studies have been conducted on ATS data sets, methods, and techniques, there is still a lack of methods that can generate comprehensive summaries of text. This paper aims to fill this gap by developing a text summarization method based on fuzzy logic using fuzzy rules in fuzzy inference systems. Additionally, the proposed method employs the TF-IDF text analysis approach with fuzzy logic, where fuzzy rules are used to select the most important sentences for the summary. The method is evaluated using an experimental analysis of a curated text dataset. The results show that the proposed method generates complete and effective text summaries, outperforming similar methods.**

I. Introduction

In the age of the information explosion, the availability of large amounts of textual data has made it difficult to process and understand this information efficiently. Text summarization has emerged as a useful solution for dealing with large amounts of text by providing users with a concise and consistent summary that captures the essence of the original content [1]. Automatic text summarization aims to generate concise summaries that capture the original text's key sentences and relevant information. The two most common summarization techniques are extractive and abstractive summarization. Extractive summarization extracts sentences directly from the source text, retaining the wording and structure of the original sentences. It relies on indicative keywords to identify important sentences, while abstractive summarization generates new sentences by rewriting and paraphrasing the content. Text summarization has many applications. A news summary service can give readers a quick overview of current events. Researchers can use this overview to sift through large numbers of research papers quickly. Additionally, chatbots and virtual assistants often use summaries to answer user queries concisely.

This paper aims to develop an extractive text summarization model using a fuzzy-based text summarization technique. Many previous works have used text analysis techniques such as TF-IDF, BoW, and other machine-learning techniques. However, the use of fuzzy-based methods for text summarization is rarely explored before. The presented text summarization model using fuzzy rules involves the application of fuzzy logic principles to condense lengthy pieces of text while retaining essential information. Fuzzy rules represent imprecise and uncertain linguistic relationships, making them suitable for capturing the nuances in human language. By assigning degrees of membership to various linguistic terms, such as "important," "relevant," or "unimportant," fuzzy logic enables the creation of summarization rules that accommodate the inherent subjectivity of summarization. These rules then guide the process of selecting and prioritizing sentences or phrases in the original text, resulting in a concise and coherent summary that captures the main ideas and key points with greater sensitivity to the subtleties of language than traditional methods. In this paper, we first implemented the model based on a fuzzy inference system or, more precisely, fuzzy rules to prioritize the sentences. Then, we applied another method involving feature weights rather than fuzzy rules for calculating the sentence scores. Additionally, we employed a genetic algorithm to optimize feature weights used in the model. Finally, we applied the developed model to the curated text dataset to evaluate its performance for text summarization. The results showed that the proposed method generates complete and effective text summaries, outperforming similar methods.

The rest of the paper is organized as follows. .....

II. Literature Survey

In the field of text summarization based on fuzzy logic, several authors have examined a range of techniques and methods to enhance automatic summarization processes. Adhika Pramita Widyassari and supriadi Rustad (2022) conducted a re-

view encompassing both abstractive and extractive approaches to automatic text summarization, presenting a comprehensive compilation of existing studies. Similarly, Wafaa S. El-Kassas and Cherif R. Salama (2021) provided an encompassing survey of approaches to automatic text summarization. Fábio Bif Goularte and Silvia Modesto Nassar (2019) introduced a novel text summarization method rooted in fuzzy rules, which they applied to automated assessment. Their results demonstrated an improvement in f-measure compared to some naive methods. S.A. Babar and Pallavi D. Patil (2015) used fuzzy logic and semantic analysis to enhance text summarization performance. Their graphical results indicated the superiority of their proposed summarizers over traditional fuzzy summarization approaches. Ladda Suanmali, Naomie Salim, and Mohammed Salem Binwahlan (2009) presented a fuzzy logic-based method to enhance text summarization. Their results showcased the efficacy of their approach, particularly in terms of f-measure, enhancing summary quality through fuzzy logic as opposed to conventional statistical methods. Collectively, these studies contribute to the advancement of text summarization by harnessing fuzzy logic and related techniques, shedding light on methodologies, and showcasing improved summarization outcomes.

III. Fuzzy-based Model for the Extractive Text Summarization

Figure 1 provides an overview of the extractive text summarization model presented in this paper. The model takes source documents written in the English language as input. Text preprocessing techniques are applied to the input text, such as text cleaning, tokenization, stop word removal, limitization, and sentence segmentation. Next, feature extraction is performed on the text. The extracted features are then given as input to the fuzzy system, which derives fuzzy rules of the text and generates sentence scores. These scores are then used to generate extractive sentences summarizing the input text documents.

A. Data Pre-processing

The following steps are applied to pre-process the input text.

- **Text cleaning:** Remove special characters, symbols, and punctuation marks. Convert the text to lowercase for case insensitivity. Remove extra spaces and tabs.
- **Tokenization:** Break the text into individual words or tokens. This step is essential for further processing and analysis.
- **Stop word removal:** Eliminate common and non-informative words such as "the," "and," "in," "is," etc., which do not contribute significantly to the summary.
- **Lemmatization or stemming:** Reduce words to their base or root form. This helps in consolidating similar words, e.g., "running" and "ran" to "run."
- **Part-of-speech tagging (POS tagging):** Assigning grammatical tags to each word, such as noun, verb, adjective, etc. This information can be useful for certain summarization approaches but is not always required. We will use this while extracting our thematic words from the text.
- **Sentence segmentation:** Split the text into individual sentences. A summary typically consists of a selection of important sentences.

B. Defining and Extracting Text Features

After completing the preprocessing steps, every sentence in the document is converted into an attribute vector that contains various features. These features serve as attributes that represent the information relevant to their specific purpose. We focus on seven distinct features for each sentence. Each feature is assigned a numerical value ranging from 0 to 1. The seven features are as follows:

1) **Title Feature Score:** The title feature score is a measure of the similarity between a sentence (s) and the document's title. It is calculated by counting the number of times each word from the sentence appears in the title and giving a score based on how many times. To implement this feature, the sentence and the title should be tokenized into words. The score is determined by dividing the total number of words in the sentence that appear in the title by the number of words in the title.
   F1(S) = No. Title word in S / No. Word in Title
2) **Sentence Length:** This feature measures sentence length by counting the number of words in each sentence. Longer sentences are thought to carry more information than shorter sentences. This feature can be used to filter out short sentences, such as datelines and author names, which are not typically included in summaries. The procedure works by first tokenizing each sentence into words. The number of words in each sentence is then counted. The sentence length feature is calculated as the ratio of the number of words in the sentence to the number of words in the longest sentence in the document.
   F2(S) = No. Word occurring in S / No. Words occurring in the longest sentence
3) **Sentence Position:** The sentence position feature measures the importance of a sentence based on its placement in the text. This can include factors such as the sentence's location within a section or paragraph. It is suggested that the first sentence has the most priority, followed by the second sentence, and so on. For example, if the first five sentences of a paragraph are considered for this feature, the scores would be as follows. F3(S) = 5/5 for the first sentence, 4/5 for the second sentence, 3/5 for the third sentence, 2/5 for the fourth sentence, 1/5 for the fifth sentence, and 0/5 for all other sentences.
4) **Sentence Similarity:** It measures the similarity between sentences by using the cosine similarity metric. This metric outputs a value between 0 and 1, where 0 indicates no similarity and 1 indicates perfect similarity. The calculation of sentence similarity score is performed as follows. Calculate the cosine similarity between each

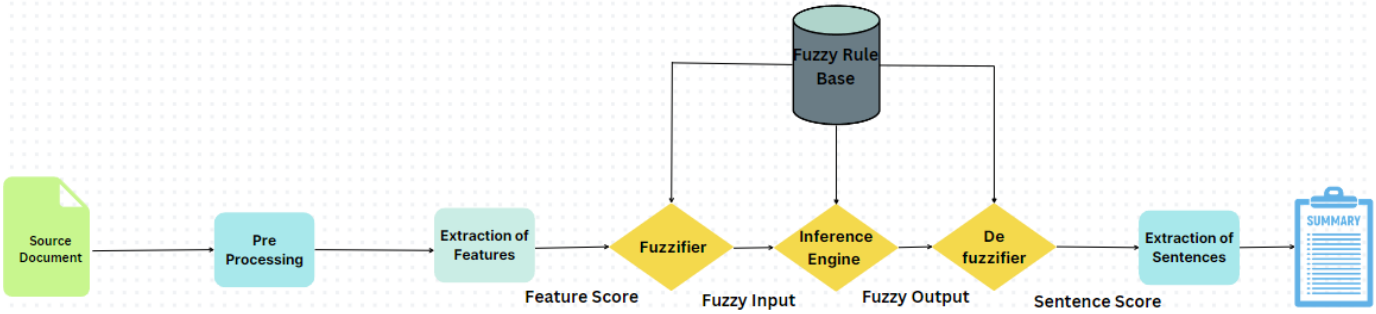

Fig. 1. Overview of the presented text summarization model

sentence and all other sentences in the document. Take the average of the cosine similarities for each sentence. The sentence similarity score for each sentence is the ratio of its average cosine similarity to the maximum average cosine similarity in the document.

5) **Numerical score ratio:** It determines the proportion of numerical data to all the words in a sentence. It can be used to identify sentences that contain statistics or numerical data. The numerical score ratio is calculated by counting the number of numerical data points (such as numbers) in a sentence and dividing it by the total number of words in the sentence.
F5(S) = No. Numerical data in S / Sentence Length (S).
6) **Thematic word feature score:** This feature calculates the score by counting the number of thematic words (such as nouns) contained in a sentence. The frequency of thematic words per sentence is an important characteristic because words that occur frequently within a document are likely related to the topic. The number of thematic words indicates the terms with the highest degree of relevance. The steps involved in calculating the thematic word score are as follows. Tokenize the sentence, remove stop words, and tag the parts of speech. Identify the thematic words in the sentence. Calculate the score as the ratio of the number of thematic words that occur in the sentence to the maximum number of thematic words in the sentence.see thematic words can be extracted from the text as well as title of the text .
F6(S) = No. Thematic word in S/Max No. Thematic word
7) **Sentence Weight:** It is calculated using the TF-IDF technique .

C. Input Text Representation

The user needs to enter the title,the number of lines of summary they need and the text paragraph for which he wants summary. The entered title will help in calculating the title feature score . The entered text can be of different pages or paragraphs. However, we will convert the whole into a single paragraph to proceed.

D. Fuzzy-based Model Development

Fuzzification converts numerical scores into linguistic terms using a triangular membership function. Various fuzzy membership functions are available, and in this work, we used triangular ones to make our calculations faster and more precise. Fuzzy rules are then applied to determine the importance level of each sentence. Defuzzification transforms the fuzzy output into a categorical representation to categorize sentences as unimportant, average, or important.

1) Fuzzification:: Fuzzification is the process of mapping numerical values to linguistic terms or fuzzy sets. In this case, we use a triangular membership function for fuzzification. A triangular membership function is a common choice because it allows for a gradual transition between linguistic terms. Each score, which represents the importance of a sentence, is fuzzified using the triangular membership function. The triangular membership function defines three linguistic terms: LOW, MEDIUM, and HIGH. These terms are represented by triangular-shaped fuzzy sets.

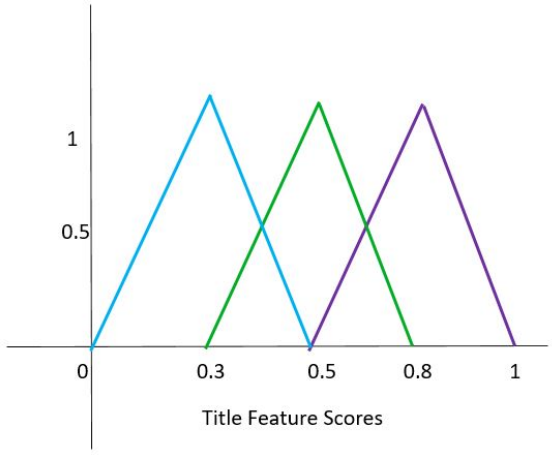

Fig. 2. Fuzziication process

Let's assume we have a sentence with a sentence length score of 0.3. To fuzzify this score, we calculate the degree of membership for each linguistic term using the triangular membership functions.

- For the LOW term: Degree of membership = 1 - (0.3 - 0)/(0.4 - 0) = 0.75.
- For the MEDIUM term: Degree of membership = (0.3 - 0.2)/(0.4 - 0.2) = 0.5.
- For the HIGH term: Degree of membership = (0.3 - 0.2)/(0.4 - 0.2) = 0.5.

So, the fuzzified values for this sentence's length would be: LOW: 0.75, MEDIUM: 0.5, HIGH: 0.5. This process is repeated for all sentences, fuzzifying their length scores into the corresponding linguistic terms. Further, we will calculate fuzzy values for each feature like this corresponding to each sentence.

We used fuzzy rules and fuzzy values together to calculate the consequence based on the principles of fuzzy logic. Fuzzy rules define the relationships between the linguistic variables of the features, and fuzzy values represent the degree of membership of each linguistic variable for a particular feature in a sentence. The function determines the most appropriate consequence label for the given sentence by evaluating the fuzzy rules with the fuzzy values.

2) Rule Base: Once the scores have been fuzzified into linguistic terms, fuzzy rules are applied to determine the importance of each sentence. Fuzzy rules are predefined statements that relate the input (linguistic terms) to the output (importance level) based on expert knowledge or predefined heuristics. The fuzzy rules can take various forms and depend on the specific context and requirements of the text summarization system. These rules can be defined using IF-THEN statements or other logical expressions. For instance, a fuzzy rule might state:

IF (NoWordInTitle is VH) and (SentenceLength is H) and (TermFreq is VH) and (SentencePosition is H) and (SentenceSimilarity is VH) and (NoProperNoun is H) and (NoThematicWord is VH) and (NumbericalData is H) THEN (Sentence is important)

By applying the fuzzy rules to the fuzzified scores, the system assigns an importance level (unimportant, average, or important) to each sentence.

3) Fuzzy Inference: After applying the fuzzy rules to determine the overall importance of each sentence based on the fuzzy inputs, we used fuzzy inference methods such as Mamdani or Sugeno to aggregate the fuzzy outputs. In the fuzzy inference system, we have used the c*alculateConsequence* function to determine the output/consequence of each fuzzy rule, which works as follows: The *fuzzyRules* parameter represents a list of fuzzy rules. A fuzzy rule consists of an antecedent and a consequence. The antecedent describes the conditions or inputs, while the consequence represents the output or result. The *fuzzyValues* parameter is a dictionary that maps fuzzy variables to their corresponding membership values. These membership values indicate a linguistic variable's degree of membership or truthfulness for a particular feature. The function takes the generated fuzzy rules (*fuzzyRules*) and fuzzy values (*fuzzyValues*) as inputs. It initializes an empty dictionary called *consequenceValues*. This dictionary will store the calculated consequence values for each fuzzy rule. It iterates over each fuzzy rule in the *fuzzyRules* list. Each fuzzy rule evaluates the antecedents by extracting the feature and linguistic value from each antecedent string. It looks up the corresponding fuzzy value from the *fuzzyValues* dictionary. It checks if all the antecedent results are greater than 0. If they are, it means the fuzzy rule matches the fuzzy values. If the fuzzy rule matches, it retrieves the consequence from the fuzzy rule and calculates the minimum of the antecedent results. It updates the *consequenceValues* dictionary with the calculated consequence value. If the consequence already exists in the dictionary, it takes the maximum value between the existing value and the new value. After evaluating all the fuzzy rules, it checks if there are any calculated consequence values in the *consequenceValues* dictionary. If there are, it finds the maximum consequence value and retrieves the corresponding consequence which will act as a sentence score. Finally, it returns the determined consequence. If none of the fuzzy rules match, it returns the default consequence of Unknown.

4) Defuzzification: To convert the fuzzy output into a crisp value representing the degree of importance for each sentence, we can use defuzzification techniques such as centroid, mean of maxima, or weighted average. However, we have just calculated the importance of each sentence using the consequence value returned by our function. This value will act as a sentence score, and we have not performed defuzzification in the net. Sentence selection is performed by ranking the sentences based on their importance scores obtained from defuzzification. The top-ranked sentences are then selected to form the summary. The selected sentences are concatenated in the post-processing step to generate the final summary.

E. Text Summarization using Feature weights by Fuzzy

In the presented model, a *featureWeights* dictionary is prepared that assigns weights to different features used to calculate the scores for each sentence. The purpose of these weights is to control the importance of each feature in the overall sentence ranking process. Domain knowledge or the optimization of the feature weights can also be applied to select the optimal feature weights.

The Title Feature Scores and Thematic Word Feature Scores are given approximately equal weights, suggesting that the model considers both the alignment with the main title and the presence of thematic words in sentences as somewhat equally important in determining sentence scores. We have assigned a higher weight to sentence similarities, indicating that the similarity of a sentence to other sentences in the text is considered more critical than the position of the sentence within the text. This is basically driven by the assumption that sentences that are more similar to others are more likely to contain essential information and capture the core concepts of the text. Sentence lengths are considered more significant than numerical Score Ratios because we need small-sized summaries. However, when we are extracting important points from a text, both the features can have the same weight, implying that the model perceives sentence length and numerical score ratios as equally significant.

We have used the max membership method to calculate the sentence score (for a single sentence). This method extracts the maximum membership value among the low, medium, or high sets for a particular feature corresponding to a sentence. The extracted membership value is then multiplied by the respective feature weight. These two steps are applied to each feature of the given sentence. Finally, the final values for all the features are added together to calculate the overall sentence score. After finishing each sentence's score calculation, we performed sentence selection similar to the fuzzy inference method. This means we will simply select the top-ranked sentences to form a summary.

F. Fuzzy-weight based methods for text extractive summarization

The fuzzy rules method uses predefined rules for summarization based on sentence features. The fuzzy weighted method, on the other hand, uses fuzzy logic to score sentences by handling feature uncertainty. In the fuzzy weighted method, feature values are fuzzified and then combined to obtain overall scores. To optimize the feature weights, we used a genetic algorithm. We used the real-coded genetic algorithm, not the binary one, because we have real values. We don't want to convert them into binary and then back into decimal values again, as we need the real-coded values to calculate the fitness. Also, if we use binary code, we will have long string lengths and will face the problem of Hamming cliffs, where we need to change a lot of bits just to go to the next value for a particular feature.

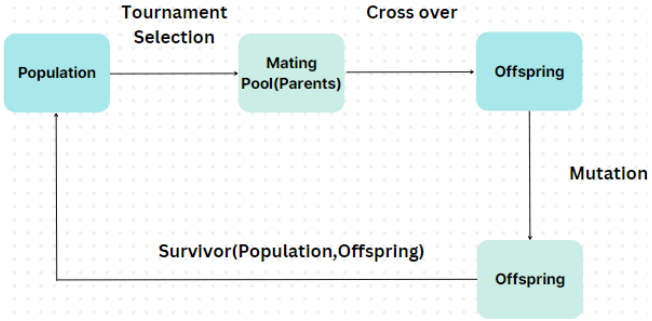

Fig. 3. Genetic algorithm-based fuzzy weight optimization

We have employed the f-score as our fitness function, which compares a predetermined gold summary with a model summary. The model summary is generated iteratively by optimizing feature values. Population initialization can be random or heuristic-driven, as we know about feature importance. We have used the following methods for each step:

- **Parent selection:** A 2-tournament selection mechanism is used to select individuals from the current population to become parents for the next generation. It forms the mating pool. The basic idea is to randomly select a small subset of individuals (the tournament) from the population and then choose the best individual from that subset to be a parent. This process is repeated until a sufficient number of parents are selected for the next generation. The key parameter in tournament selection is the tournament size, which determines how many individuals participate in each tournament. In this case, we use a tournament size of 2.
- **Crossover:** Simulated binary crossover (SBX) is used because naive crossover operators, such as the single-point crossover, may fail to perform well. This is because naive crossover operators only search within the current values of the decision variables. As a result, they eventually depend on the mutation operator to generate new values for the decision variables. This is why simulated binary crossover is preferred. SBX simulates the single-point crossover on binary strings. It requires two parents to generate two offspring. The two offspring have a separation proportional to the parents. Crossover is performed with a high probability.
- **Mutation:** The algorithm uses environment selection. We combine the generated offspring from the mutation and initial population and then arrange them according to their fitness values. Since our problem is maximization, we will select the top individuals, the same size as the initial population. These selected individuals will then become the new initial population of the next generation. This will help us to optimize our features and values.

Algorithm-1 and Algorithm-2 describe the process of cross-over and mutation used in the fuzzy-weight-based method.

Algorithm 1: Procedure for SBX cross over

Input: P, D, P_c, and η_c ;
where, P = Population, D = Dimension of the data, P_C = Cross-over probability, η_C = cross-over distribution

begin
  Randomly select pair of parents(P'_a,P'_b) from mating pool. ;
  Generate a random number(r) between 0 and 1. ;
  if $r \geq p_c$ **then**
    Copy the parent solutions as offspring
  end if
  if $r < p_c$ **then**
    Generate D random numbers (u) for each variable.
  end if
  Determine β of each variable. ;
  Generate two offspring (O_a and O_b) using
  O_a =0.5[(1+β)$P'_a + (1-\beta)P'_b$]
  O_b =0.5[(1-β)$P'_a + (1+\beta)P'_b$]
end

IV. Experiments and Results

We used the F-score measure to evaluate the performance of the presented model. The F-score is a combination of precision and recall, and provides a balanced measure of the summarization system's performance. It is defined as follows.
1. True Positives (TP): The number of words (such as sentences or phrases) in the reference summary that are also present in the generated summary.
2. False Positives (FP): The number of words in the generated summary that are not present in the reference summary.

Algorithm 2: Procedure for polynomial Mutation

Input: P, D, P_m, and η_m

begin

  Generate a random number (u) between 0 and 1 ;

  if $u \geq p_m$ **then**

    No change in the offspring

  end if

  if $u < p_m$ **then**

    generate D random numbers (r) corresponding to each variable

  end if

  Determine δ of each variable

$$\delta = \begin{cases} (2r)^{\frac{1}{\eta_m+1}} - 1 & \text{if } r < 0.5 \\ 1 - [2(1-r)]^{\frac{1}{\eta_m+1}} & \text{if } r \geq 0.5 \end{cases}$$

  Modify Off springs using o = o + $(x^u - X^l) \cdot \delta$

end

3. False Negatives (FN): The number of words in the reference summary that are missing from the generated summary.
4. Precision: It measures how many generated words are actually relevant or correct, out of the total generated words. It is calculated as given in Eq-1.

$$Precision = \frac{TP}{TP + FP} \tag{1}$$

5. Recall: It measures how many relevant or correct words from the reference summary were captured by the generated summary. It is calculated as given in Eq-2.

$$Recall = \frac{TP}{TP + FN} \tag{2}$$

6. F-score: It is the harmonic mean of precision and recall. It is calculated as defined in Eq-3.

$$F - score = \frac{2 * (Precision * Recall)}{Precision + Recall} \tag{3}$$

A. Results and Analysis

We curated the dataset to calculate the performance of the presented fuzzy-based model. The curated dataset contains a reference summary (also called a gold summary) and the generated summary produced by our summarization system. We compared the content of the generated summary to the reference summary to count the number of true positives, false positives, and false negatives. We then used these counts to calculate the precision, recall, and F-score. Our dataset is a collection of random 20-line online paragraphs. Their human-generated summaries will serve as gold summaries. Table I shows the F-scores for the three models discussed in the paper. As we can see, the fuzzy inference model using fuzzy rules is much better than the other two. Although we have optimized the fuzzy method using feature weights, we cannot achieve the maximum F-score due to early convergence.

Figure 4 shows the output of the presented model applied to the real user input. The left part of the figure shows the input area, and the right part shows the generated output summary. It can be observed that the presented model can effectively generate the summary of the user input text.

TABLE I
RESULTS OF THE PRESENTED FUZZY-BASED MODELS FOR EXTRACTIVE TEXT SUMMARIZATION IN TERMS OF F-SCORE

Measure	TF-IDF	Fuzzy (using feature weights)	Fuzzy (using fuzzy rules)
Sample-1	0.20	0.42	0.44
Sample-2	0.54	0.56	0.66
Sample-3	0.24	0.42	0.58

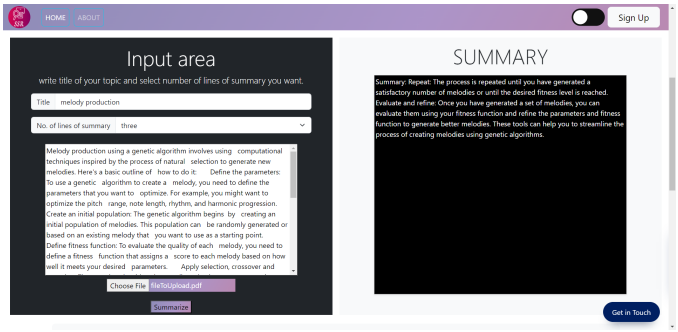

Fig. 4. Extractive Summarization

B. Discussion

The presented fuzzy-based models produced moderate results for the extractive text summarization. Text summarization based on fuzzy rules can be improved in several ways. We can add new features, assign appropriate membership functions, define different linguistic variable sets, and optimize the boundaries of the membership functions. We can also improve text summarization based on feature weights by optimizing the feature weights using the same methods discussed above. Additionally, we can fine-tune the model by assigning different weights to various features to produce summaries that align with the desired characteristics or priorities of the summarization process.

V. CONCLUSION

The fuzzy system method for text summarization offers a valuable and effective approach for processing and summarizing information. The summarization process becomes more adaptive and flexible by leveraging fuzzy logic to handle linguistic uncertainties and approximate reasoning. The presented method successfully captures the essence of the original text by extracting key features and linguistic patterns and generating a concise summary that retains essential information. The fuzzy system's ability to handle vague and imprecise data makes it suitable for text summarization tasks, as natural language is inherently ambiguous. The method's performance can be optimized through appropriate parameter tuning and linguistic rule refinement, resulting in even more accurate and informative summaries. Depending solely on fuzzy logic might have limitations in capturing very subtle nuances or complex relations present in some texts. Therefore, researchers and developers should continue exploring hybrid approaches that combine fuzzy systems with other AI techniques to achieve

more comprehensive and refined summaries. Overall, the fuzzy system method is a valuable tool for text summarization. Further advancements in this area could lead to significant improvements in automatic summarization algorithms, which would benefit various applications in information retrieval, natural language processing, and knowledge extraction.

REFERENCES

[1] W. S. El-Kassas, C. R. Salama, A. A. Rafea, and H. K. Mohamed, "Automatic text summarization: A comprehensive survey," *Expert systems with applications*, vol. 165, p. 113679, 2021.