Journal of Data Acquisition and Processing
ISSN: 1004-9037
https://sjcjycl.cn/
DOI: 10.5281/zenodo.98549721

MARATHI EXTRACTIVE TEXT SUMMARIZATION USING LATENT SEMANTIC ANALYSIS AND FUZZY ALGORITHMS

Virat V Giri
Principal, Sanjay Ghodawat Polytechnic, Kolhapur, India.

Dr. M.M. Math
Professor, Dept of CSE, KLS, Gogte Institute of Technology, Belgaum, India.

Dr. U.P. Kulkarni
Professor, Dept. of CSE, SDM CET, Dharwad, India.

***Abstract*—**Extractive text summarization involves the retention of only the most important sentences in a document. In the past, multiple approaches involving both statistical and machine learning-based methods have been used for this task. The crucial step in extractive text summarization is getting the right ranking order of sentences in the document in terms of their importance. Singular value decomposition or SVD algorithm based on latent semantic analysis focuses on recognizing the sections in the document which are related in terms of their semantic nature. Fuzzy algorithms involve reasoning of the priority order of the sentences using fuzzy logic unlike the use of discrete values. While significant work has been done for extractive text summarization in English and other foreign languages, there is ample scope for improving the performance of systems when dealing with Marathi text. In this paper, SVD and fuzzy algorithms are proposed for performing extractive text summarization on Marathi documents. Work is done upon the modeling principle, data flow, and parameters of these algorithms such that they are best suited for the task. An analysis of the characteristics of both these techniques is conducted to compare their benefits and shortcomings. The performance of both the algorithms is evaluated on a document dataset using standard performance metrics including the ROUGE metric. An unbiased comparison of both these techniques is carried out to inform the applicability of them, especially when working with Marathi or in general, non-English text.
***Index Terms*—**Extractive text summarization, Singular value decomposition, fuzzy logic, Marathi text

I. INTRODUCTION

Natural language processing involves processing and modeling of natural language data to improve understanding of computers while ensuring that the semantic and syntactic structure of the data is retained. Text summarization is an important application of natural language processing which focuses on automatically deriving the summary of entered documents. There are two possible types of text summarization: abstractive text summarization and extractive text summarization.

Extractive text summarization is a summarization type where there is no addition of new content or modification of existing content, but rather only the most important phrases and

sentences from the document are retained as the summary of document [1], [2]. It is akin to a highlighter used to highlight only the most important sections of a document to the viewer. Such an algorithm would require an accurate ranking of the sentences present in the document based on their relevance to the summary. Based on a decided threshold, the top *N* ranked sentences would then be predicted as the summary of the document. Previous approaches in this domain have considered the use of statistical features such as word count, and term frequencies for ranking the sentences [3]. Traditionally, extractive text summarization algorithms have been demonstrated and conceptualized while taking into consideration the English language [4]. Recent years have seen a rise in the contributions from the research community for non-English languages, including many Indian languages [2]. Marathi is a language spoken in the state of Maharashtra and nearby regions in India and is derived from the Devanagari script [2]. The work on extractive text summarization in the Marathi language has been relatively paltry [5]–[7].

In this paper, two different approaches are proposed for performing extractive text summarization on Marathi text documents. The first approach focuses upon using latent semantic analysis (LSA) that deploys a semantic identification and correlation of sentences between a document [8]. This is achieved using the singular value decomposition (SVD) technique. The second approach makes use of fuzzy logic to solve this task. The fuzzy algorithm focuses on certain text characteristics and rules [9] using which sentence scores are assigned.

The contributions made through this paper are enlisted as follows:

1) Implementation of a latent semantic analysis based algorithm for extractive text summarization on Marathi documents
2) Implementation of fuzzy logic to derive rules useful for sentence ranking for extractive text summarization of Marathi documents.
3) Analysis and comparison of the aforementioned two approaches to better guide further work in this domain.

The rest of the content is structured as follows: previous work in this domain is discussed in Section 2, the two proposed approaches are described in detail in Section 3, the dataset description is detailed in Section 4, the results and analysis of the two approaches based on the results are carried out in Section 5 and the conclusion and future scope are mentioned in Section 6.

II. BACKGROUND AND RELATED WORK

Extractive text summarization has been the preliminary text summarization demonstrating the priority ranking capability of the data modeling algorithm. Previous work in this domain has focused on the use of statistical features in the early days, and recently has seen more focus on graph-based and deep learning-based methods.

The initial method for extractive text summarization made use of term frequency and inverse document frequency for feature selection [3]. Documents, however, could also include various themes that are addressed, and clustering these methods together was also used as an approach for priority ranking [10]. Kumar et al. [11] made use of a knowledge induced graph for

performing single document summarization. In the last decade, machine learning has also been used to tackle this task across multiple domains as well as multiple languages [12], [13]. Query-based text summarization has also been tried where the ranking is based on the overlapping between the query phase and the document terms [14]. Recent years have also seen attempts to perform extractive text summarization using deep learning.

Recently, there have been attempts to perform extractive text summarization in Marathi. Bhosale et al. [7] used a naive frequency count approach for this task. Rathod made use of the page-rank and text-rank algorithms, however, their evaluation of these approaches was very restricted [15]. Sarwadnya and Sonawane went a step further in terms of the use of preprocessing methods and reliance on the text-rank algorithm [5]. Chaudhari et al. [6] presented the use of deep learning to create the summarizer.

These approaches have either not been evaluated on a standard-sized dataset, or have made use of traditional approaches only. While, singular value decomposition and fuzzy logic have been explored in the English language [1], [16], there has been no significant contribution by the community for using these on Marathi documents. In this paper, an attempt is made to try to incorporate these approaches in a novel way to boost performance when working with Marathi documents.

III. PROPOSED METHODOLOGIES

Two different methods are proposed and analyzed in this paper- the first one is the use of singular value decomposition (SVD) strategy as a part of the latent semantic analysis (LSA) approach, while the second is the use of fuzzy logic and rules for deciding the ranking mechanism. Figure 1 shows the block diagram of the extractive text

Fig. 1: Block diagram of the extractive text summarization architecture

summarization architecture. Note that two different solutions are presented for the summarizer algorithm phase.

A. Preprocessing

Preprocessing steps of the input documents in both the approaches remain the same. The document is first tokenized into separate tokens. This is followed by the removal of stop words. Stop words are the words that do not add to the meaning of the sentence and are used only to ensure the grammatical consistency of the sentence. These words do not add value in terms of realizing the ranking order of the sentences as they have a uniform probability of occurring in both important and unimportant sentences.

Marathi language is characterized by the addition of suffixes to verbs to indicate the gender or the tense in which the sentence is being spoken. These suffixes also do not add any value to the semantic meaning of the sentence. They are removed to bring about faster processing and modeling and also reduce the number of distinct tokens modeled by the algorithm, thereby ensuring no ambiguous interpretations of similar meaning words. The preprocessed text is now more model-friendly and is passed as input to the summarizer algorithm.

B. Summarizer algorithms

Two different algorithms are presented in this paper for extractive text summarization. These are as follows:

1) Singular Value Decomposition (SVD): Computing the latent semantic structure of the document to obtain context similarity between the sentences and thereby mapping the vector space.
2) Fuzzy logic: Calculating values of some handcrafted statistical features and defining rules based on these features that are then passed as inputs to the fuzzy algorithm.

The inner working of both the algorithms is discussed in detail in the further subsections:

1) Singular Value Decomposition: Singular Value Decomposition or SVD is a technique under latent semantic analysis that tries to correlate and find the relation between the sentences present in a document and the words present in that sentence. The approach works in two distinct phases:

In the first phase, the input matrix D is created based on the term frequency of the words present in the document [17]. For m distinct words and n sentences in the document, D would be a mxn dimension matrix. As every word does not occur in each of the sentences, A tends to be a sparse matrix in nature. Further, every sentence row in this matrix is normalized to a range between 0 and 1 using the following equation:

$$sentence\_row = \frac{sentence\_row}{max(sentence\_matrix.value())} \quad (1)$$

Such a normalized input matrix can now be passed as an input to the SVD approach, which can be represented mathematically as follows:

$$D = U\Sigma V^T \quad (2)$$

where D: Normalized input representation matrix U: *mxn* matrix representing left singular vectors in the form of words x concept
Σ: *nxn* diagonal matrix indicating the singular
eigenvalues, descending across the diagonal *V* : *nxn* matrix indicating the right singular vectors in the form of sentence x concept

Algorithm 1 indicates the procedure to derive the SVD values for subsequent ranking of the sentences in the document

Algorithm 1 Algorithm for computing SVD

Input: Normalized input representation matrix *D*
Output: Values of *U*, *V* , and Σ
1: Prod_D = DD^T
2: x1 = Eigen_values(Prod_D)
3: Inv_D = D^TD
4: x2 = Eigen_values(Inv_D)
5: Val = $\sqrt{x1} \cap x2$
6: Assign values to *U*, *V* , and Σ
7: **return** *U*, *V* , Σ

As a modification to the existing SVD approach, three other factors are also considered. Apart from the sentence similarity weight, the sentence length, sentence position, and sentence value are also included to decide the final ranking for the summarization. Each of these factors is considered and evaluated as follows:

- Sentence length: If the sentence length is less than the minimum permissible length, or greater than the maximum permissible length, then set it to zero. Otherwise, calculate as follows:

(3)

$$Sentance\ Length = \sin\left(\frac{180 * (Length - \min\ \_length)}{(\max\ \_length - \min\ \_length)}\right)$$

• Sentence value: The normalized input representation discussed earlier

- Sentence position: If the sentence is the first or the last one in the document, then consider it to be important and set value as 1. Otherwise, derive the value as follows:

$$Sentence\ pos = cos(\frac{Pos - (TRSH * len(sentences)) * 360}{(1 - 2 * TRSH) len(sentences)}) \quad (4)$$

where *TRSH* is a hyperparameter decided by the user. The value is set to 0.01 in the presented setup.

- Sentence weight similarity: Calculated using the number of overlapping words present between two sentences.

The final ranking for the sentence is derived by considering the sum of the absolute values of each of these factors. Based on the summary factor given by the user, the ranked sentences are sorted in descending order and the filtered sentences are output as the summary of the document.

2) Fuzzy Logic: The proposed fuzzy logic is calculated using a feature matrix. The feature matrix is derived based upon certain statistical features present in the document. Each of these features is as follows:

- Position factor of the sentence: The position factor of the sentence is calculated by normalizing its order in the document with respect to the total number of sentences.

$$Pos\ factor = \frac{Total\_sentences - current\_pos}{Total\_sentences} \quad (5)$$

- Bigram token length: Bigram is the tokenization of words done, but by considering two words at a time. The number of such bigram tokens present in a sentence is considered.

Trigram token length: Trigrams are similar to bigram, but they consider three words together at a time. Trigram token length refers to the number of such trigram tokens present in the sentence.

- TF-ISF vector: It considers the term frequency as well as sentence frequency and is calculated as follows:

$$tf\_isf = \frac{term\_freq}{sent\_freq * vocab\_pos} \quad (6)$$

- Cosine similarity: Calculate the cosine similarity of the sentence with respect to the centroid of the document. Mathematically, this can be represented as follows:

$$Cos\_similarity(S, Z) = \frac{S.Z}{||S||^2.||Z||^2} \quad (7)$$

where *Z* is the centroid of the document and *S* is the sentence in consideration.

- Thematic number: It takes into consideration the factor of the number of keywords present in a sentence with respect to the total keywords present in the document [9].

$$Thematic\_number(S) = \frac{keywords\ in\ S}{total\ keywords} \quad (8)$$

- Sentence length factor: It is calculated by taking the ratio of the length of the sentence to the length of the longest sentence present in the document [18].
- Numeric tokens: The number of numeric tokens present in the sentence in consideration with respect to its length.
- Pnoun score: The ratio of the number of proper nouns present in the sentence to the total words present in it. More important sentences generally tend to contain more information which would also be proportional to the number of proper nouns present in the sentence.

For each of these fuzzy variable factors, three values (poor, average, good) are used to auto-populate them. The fuzzy logic requires a triangular membership function generator that accepts an independent variable and a three element vector used to control the shape of the function [19]. Based on the previously mentioned nine factors, a consequent factor *sent* is determined that is termed as bad, average, and good for vector values of [0,0,50], [0,50,100], and [50,100,100] respectively. Using all of this information, five rules are set to compute the fuzzy logic prediction values. The rules are as follows:

1) sent['good'] = Position factor['good'] & Sentence length['good'] & Pnoun score['good'] & Numeric tokens['good']
2) sent['bad'] = Position factor['poor'] & Sentence length['poor'] & Numeric tokens['poor']
3) sent['bad'] = Pnoun score['poor'] & Thematic number['average']
4) sent['good'] = Cosine similarity['good']
5) sent['avg'] = Bigram token['good'] & Trigram token['good'] & Numeric tokens['average'] | TF-ISF['average']

For an instance of data, the values of the aforementioned nine factors are calculated per sentence and passed as input for the fuzzy logic to compute. If the output of the consequent factor is greater than 50, the sentence is included in the summary of the document.
Using these two methods, summarization of a standard size document dataset is carried out and the obtained results are discussed in the next section.

IV. Dataset description

The performance of the two proposed approaches is evaluated on a custom created dataset consisting of Marathi news articles ranging on a diverse set of issues including politics, economics, and social affairs. The dataset consists of 100 documents coupled with their manual summaries used later for evaluation purposes. A sample instance from a document in the dataset is shown in Table I.

आर्थिक वर्षाच्या पहिल्याच दिवशी शुक्रवारी मुंबई शेअर बाजाराच्या निर्देशांकात (सेन्सेक्स) 72 अंशांची घसरण होऊन तो 25 हजार 269 अंशांवर बंद झाला. आशिया आणि युरोपीय शेअर बाजारातील घसरणीमुळे हा परिणाम झाला. दरम्यान, राष्ट्रीय शेअर बाजाराच्या निर्देशांकात (निफ्टी) 25 अंशांनी घसरून 7 हजार 713 अंशांवर बंद झाला. देशातील मोटारनिर्मिती क्षेत्रातील आघाडीची कंपनी मारुती सुझुकीच्या मार्च महिन्यातील विक्रीत 15.9 टक्के वाढ झाल्याने कंपनीच्या समभागात आज 0.11 टक्के वाढ नोंदविण्यात आली. कर्जाच्या विळख्यात सापडलेल्या जयप्रकाश असोसिएट्सने सिमेंट व्यवसायातील काही हिस्सा कुमारमंगलम बिर्ला यांच्या मालकीच्या अल्ट्रा टेक कंपनीला 15 हजार 900 कोटी रुपयांना विकण्याची घोषणा केली आहे. यामुळे जयप्रकाश continue

TABLE I: Sample document text

As both the methods are instance-based and do not involve any trainable parameters, the entire dataset consisting of all the 100 documents is used for evaluation purposes. The algorithm is predefined and hence segregation of data is not required with only one pipeline required for the entire task.

V. RESULT AND ANALYSIS

Extractive text summarization focuses on retention of the most important sections of the document. As a result, evaluation of such summarizers focuses on the amount of overlap between the human summary and the machine generated summary. To define this measure of overlap in a standard format, the ROUGE metric is used [20].

Given a human generated summary *H* and a machine generated summary *M*, the precision, recall, and the F1 score is defined as follows:

$$ROUGE1\ Precision = \frac{H \cap M}{M} \tag{9}$$

$$ROUGE1\ Recall = \frac{H \cap M}{H} \tag{10}$$

$$ROUGE1\ F1 = \frac{2 * R1.Precision * R1.Recall}{R1.Precision + R1.Recall} \tag{11}$$

Where *ROUGE*−1 refers to the overlap when considering unigrams i.e. one token at a time. In a similar manner, ROUGE-2 related metrics can be defined as follows:

$$ROUGE2\ Precision = \frac{bigrams\ in\ H \cap bigrams\ in\ M}{bigrams\ in\ M} \tag{12}$$

$$ROUGE2\ Recall = \frac{bigrams\ in\ H \cap bigrams\ in\ M}{bigrams\ in\ H} \tag{13}$$

The ROUGE2 F1 score is the harmonic mean of precision and recall. The ROUGE-L metric is defined similarly and refers to the longest matching subsequence amongst the two summaries [20]. Firstly, the performance of the approaches is evaluated on single-document summarization. Based on the mentioned performance metrics, the results are produced and tabulated in Table II.

Metric	**SVD**	**Fuzzy Logic**
ROUGE1:Precision	0.632	**0.641**
ROUGE1:Recall	0.623	**0.625**
ROUGE1:F1	0.612	**0.623**
ROUGE2:Precision	0.531	**0.561**
ROUGE2:Recall	0.519	**0.546**
ROUGE2:F1	0.512	**0.546**

ROUGEL:Precision	**0.665**	0.659
ROUGEL:Recall	0.614	**0.636**
ROUGEL:F1	0.626	**0.64**

TABLE II: Results obtained on both the approaches for single document summarization

It can be seen that the Fuzzy logic turns out to be a better approach as compared to the SVD method with better results on almost all of the performance metrics. Next, multi-document summarization is considered. In this case, the overlaps of tokens in the human summary and machine-generated summary is considered across multiple documents. Evaluation is done for precision, recall, and the F1-score. The results obtained are shown in

Table III.

Metric	**SVD**	Fuzzy Logic
Precision	**0.705**	0.625
Recall	**0.693**	0.655
F1	**0.682**	0.63

TABLE III: Results obtained on both the approaches for multi-document summarization

It can be seen that while SVD was lagging in case of single document summarization, it outperforms Fuzzy logic by a comfortable margin when it comes to multidocument summarization. Further, as a part of ablation studies, the F1 scores are compared with a recently used method of position-based textrank [21].
While position-based rank seems to be the better performer for single document summarization, SVD turns out to be the better choice when working with multidocument summarization on Marathi documents. The results obtained for both the approaches on single-document

Metric	**Position based Textrank**	**SVD**	**Fuzzy Logic**
Multidocument F1	0.667	**0.682**	0.63
ROUGE1:F1	**0.646**	0.612	0.623
ROUGE2:F1	**0.592**	0.512	0.546
ROUGEL:F1	**0.659**	0.626	0.64

TABLE IV: Comparison of F1 scores with textrank algorithm and multi-document summarization are visualized in Fig. 2 and 3.

To analyze the findings, the advantages and limitations of both the proposed approaches in the case of Marathi language are noted in Table V.

Approach	Fuzzy logic	SVD
Advantages	Can model nonlinear functions of arbitrary complexity [19] - Flexible and easy to implement - Based on natural language Accommodates faulty data	Performs better on multi-document summarization Acts as an initial step to advanced dimensionality reduction methods like PCA [8] Performs satisfactory approximation of data
Shortcomings	Does not consider the semantic analysis of the words Does not consider the correlation amongst sentences The number of rules keep on increasing with the number of input features	Might underperform on non-linear data - Fails to recognize the context and the meaning of polysemic words in the particular instance Semantic analysis is not performed

TABLE V: Analysis of two presented approaches

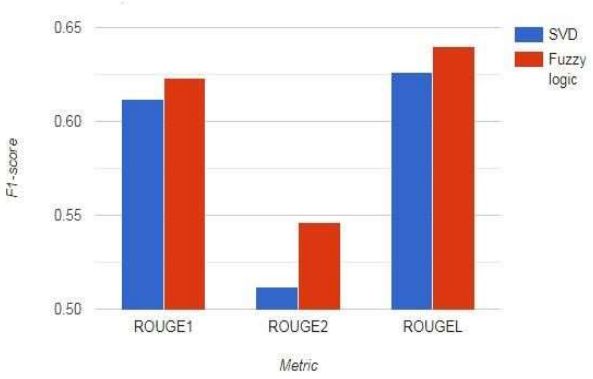

Fig. 2: Visual comparison of results obtained by both the approaches on single document summarization

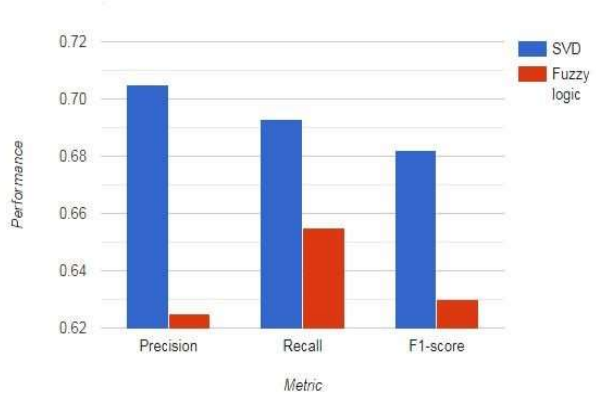

Fig. 3: Visual comparison of results obtained by both the approaches on single document summarization

VI. CONCLUSION

In this paper, two novel approaches have been proposed for the task of extractive text summarization on Marathi documents. The first approach focused on singular value decomposition as a dimensionality reduction and feature selection technique. It took into consideration the sentence position and sentence length factors along with the calculation of eigenvectors. The second approach made use of fuzzy logic to derive rules used for priority ranking of sentences based on certain statistical features in the document. Both the proposed approaches have certain advantages and shortcomings. The evaluation of the approaches was done on a standard-sized dataset and a fuzzy logic-based approach was found to be better when working on single document classification. On the other hand, SVD was seen to be the better method for multi document summarization. There is a certain accuracy complexity tradeoff amongst the two approaches. This has been demonstrated by the evaluation of multiple performance metrics. As a part of ablation studies, the results were compared with another baseline method, and due analysis was carried out. Future scope in this domain includes consideration of semantic analysis, word embeddings, and extension of the task to include abstractive text summarization. Further, code-mixed text and low resource languages can be explored for this task. The proposed approaches have shown promising signs for text summarization task in Marathi language and could be extended further to other natural language understanding tasks.

REFERENCES

[1] 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.

[2] V. V. Giri, M. Math, and U. Kulkarni, "A survey of automatic text summarization system for different regional language in india," *Bonfring International Journal of Software Engineering and Soft Computing*, vol. 6, no. Special Issue Special Issue on Advances in Computer Science and Engineering and Workshop on Big Data Analytics Editors: Dr. SB Kulkarni, Dr. UP Kulkarni, Dr. SM Joshi and JV Vadavi, pp. 52–57, 2016.

[3] J. Kaur and V. Gupta, "Effective approaches for extraction of keywords," *International Journal of Computer Science Issues (IJCSI)*, vol. 7, no. 6, p. 144, 2010.

[4] K. Kaikhah, "Text summarization using neural networks," 2004.

[5] V. V. Sarwadnya and S. S. Sonawane, "Marathi extractive text summarizer using graph based model," in *2018 Fourth International Conference on Computing Communication Control and Automation (ICCUBEA)*, pp. 1–6, IEEE, 2018.

[6] A. Chaudhari, A. Dole, and D. Kadam, "Marathi text summarization using neural networks," *International Journal for Advance Research and Development*, vol. 4, no. 11, pp. 1–3, 2019.

[7] M. S. Bhosale, D. Joshi, M. V. Bhise, and R. A. Deshmukh, "Automatic text summarization based on frequency count for marathi e-newspaper," 2018.

[8] R. M. Badry, A. S. Eldin, and D. S. Elzanfally, "Text summarization within the latent semantic analysis framework: comparative study," *International Journal of Computer Applications*, vol. 81, no. 11, pp. 40–45, 2013.

[9] J. Yadav and Y. K. Meena, "Use of fuzzy logic and wordnet for improving performance of extractive automatic text summarization," in *2016 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, pp. 2071–2077, IEEE, 2016.

[10] A. R. Deshpande and L. Lobo, "Text summarization using clustering technique," *International Journal of Engineering Trends and Technology*, vol. 4, no. 8, pp. 3348–3351, 2013.

[11] N. Kumar, K. Srinathan, and V. Varma, "A knowledge induced graph-theoretical model for extract and abstract single document summarization," in *International Conference on Intelligent Text Processing and Computational Linguistics*, pp. 408– 423, Springer, 2013.

[12] K. Sarkar, M. Nasipuri, and S. Ghose, "Using machine learning for medical document summarization," *International Journal of Database Theory and Application*, vol. 4, no. 1, pp. 31–48, 2011.

[13] M. BAZRFKAN and M. RADMANESH, "Using machine learning methods to summarize persian texts," *Indian J. Sci. Res*, vol. 7, no. 1, pp. 1325–1333, 2014.

[14] I. Imam, N. Nounou, A. Hamouda, H. Allah, and A. Khalek, "Query based arabic text summarization 1," 2013.

[15] Y. V. Rathod, "Extractive text summarization of marathi news articles," 2018.

[16] P. D. Patil and N. Kulkarni, "Text summarization using fuzzy logic," *International Journal of Innovative Research in Advanced Engineering (IJIRAE)*, vol. 1, no. 3, pp. 42–45, 2014.

[17] J. Steinberger and K. Ježek, "Text summarization and singular value decomposition," in *International Conference on Advances in Information Systems*, pp. 245–254, Springer, 2004.

[18] L. Suanmali, M. S. Binwahlan, and N. Salim, "Sentence features fusion for text summarization using fuzzy logic," in *2009 Ninth International Conference on Hybrid Intelligent Systems*, vol. 1, pp. 142–146, IEEE, 2009.

[19] F. Kyoomarsi, H. Khosravi, E. Eslami, P. K. Dehkordy, and A. Tajoddin, "Optimizing text summarization based on fuzzy logic," in *Seventh IEEE/ACIS International Conference on Computer and Information Science (icis 2008)*, pp. 347–352, IEEE, 2008.

[20] C.-Y. Lin and F. Och, "Looking for a few good metrics: Rouge and its evaluation," in *Ntcir Workshop*, 2004.

[21] R. Mihalcea and P. Tarau, "Textrank: Bringing order into text," in *Proceedings of the 2004 conference on empirical methods in natural language processing*, pp. 404–411, 2004.