

# Extractive Text Summarization for Sports Articles using Statistical Method



Sai Teja Polisetty, K Selvani Deepthi, Shaik Ameen, Ravivarma G, M Mounisha

**Abstract:** The past decade has endorsed a great rise in Artificial Intelligence. Text summarization which comes under AI has been an important research area that identifies the relevant sentences from a piece of text. By Text Summarization, we can get short and precise information by preserving the contents of the text. This paper presents an approach for generating a short and precise extractive summary for the given document of text. A statistical method for extractive text summarization of sports articles using extraction of various features is discussed in this paper. The features taken are TF-ISF, Sentence Length, Sentence Position, Sentence to Sentence cohesion, Proper noun, Pronoun. Each sentence is given a score known as the predictive score is calculated and the summary for the given document of text is given based on the predictive score or also known as the rank of the sentence. The accuracy is checked using the BBC Sports Article dataset and sports articles of various newspapers like the New York Times, CNN. The precision of 73% is acquired when compared with System Generated Summary (SGS) and manual summary, on an average.

**Keywords:** Artificial Intelligence, Cosine similarity, Natural Language Processing, System Generated Summary (SGS), Term frequency inverse sentence frequency.

## I. INTRODUCTION

Text mining can also be referred as Text analysis, which uses different Artificial Intelligence technologies. Text mining can be referred to as deriving high-quality information by drawing patterns and identifying the important keywords from the unstructured data. Text mining tasks include text classification which is the classification of the text for example genre, the sentimental analysis which tells about how the author wrote the sentences that is in which tone, document clustering.

Text mining tasks include text classification [2] which is the classification of the text for example genre, the sentimental analysis which tells about how the author wrote the sentences that is in which tone, document clustering refers to as clustering the documents using unsupervised learning, text summarization for obtaining a short and precise text. Each task is different from each other and has its own probe and methodologies. Natural Language Processing (NLP) is all about computer and human

languages interactions. A finest procedure that helps the system in reading and understanding the given text. With the rapid increase in data the text summarization manually consumes a lot of time. That resulted in development of various summarization methods. Using them, the system should deliver a précis with the main essence of the document in reduced size. Text summarization is of two types: i) extractive ii) abstractive. Firstly, the extractive text summarization where important sentences and words from the given text document are identified and those are combined into the summary in a meaningful way. Finally, in abstractive text summarization, new sentences are fabricated, and the summary is given without the loss of information which is understandable. This can be done using different methods that include statistical measures, deep learning techniques with supervised or unsupervised learning and word vector embedding [10]. The literature survey is as follows: Fabio Bif Goularte et.al (2018) [1] proposed a method using fuzzy rules to retrieve the most important sentences in a document. Fuzzy logic is applied because it can be used for sentences having ambiguity. The summaries are trained for Brazilian Portuguese texts and the summary generated is tested by the summary given by the domain experts. The author has cross-checked with other methods as well as using ROUGE measures. The f-measure obtained is 0.95 which is huge than obtained in any other method. This method works accurately with only unigrams and it doesn't work for semantic verification of the information. Taeho Jo et.al (2017) [2] proposed a method that uses the feature vector of certain features and obtains the correlation between the vectors. In previous works only the feature value is considered but, in this work, the author considers both features and its value for better performance for this KNN (K Nearest Neighbour) is used to check the correlation between the vectors. The author has used a binary classification, KNN is used which tells whether a sentence is important or not. This is way different from text categorization which uses the previous knowledge for classifying. Deepa Anand et.al (2019) [3] proposed a method for summarizing the Indian legal judgment documents for this the author has used a neural network approach that is semi-supervised.

Manuscript received on March 15, 2020.

Revised Manuscript received on March 24, 2020.

Manuscript published on March 30, 2020.

\*Correspondence Author

**Sai Teja Polisetty\***, pursuing B. Tech final year, Department of CSE, Anil Neerukonda Institute of Technology and Sciences, India,

**Dr. K. Selvani Deepthi**, Associate Professor, Department of CSE, Anil Neerukonda Institute of Technology and Sciences, India,

**Shaik Ameen**, pursuing B. Tech final year, Department of CSE, Anil Neerukonda Institute of Technology and Sciences, India,

**Ravivarma G**, pursuing B. Tech final year, Department of CSE, Anil Neerukonda Institute of Technology and Sciences, India,

**M Mounisha**, pursuing B. Tech final year, Department of CSE, Anil Neerukonda Institute of Technology and Sciences, India,

© The Authors. Published by Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP). This is an [open access](http://creativecommons.org/licenses/by-nc-nd/4.0/) article under the CC BY-NC-ND license (<http://creativecommons.org/licenses/by-nc-nd/4.0/>)

The author used two different LSTM architectures for both word embedding and sentence embedding. The author used Glove vectors and co-selection measures are used for analysis. The recall of 0.7 to 0.8 approximately is obtained which is moderate. This approach cannot be used for long sentences so sentence simplification methods should be used. Krishnaveni et.al (2017) [4] proposed a heading-based text summarization since the context can be known from the heading. Each sentence is given a rank on considering how much relevant the sentences are with the heading and the top sentences are retrieved based on the compression ratio. Since the summary is generated based on the heading there is not irrelevancy in context. The summaries are compared with that of the summaries generated from the main summariser, Ms-word summarizer.

Jingqiang Chen et.al (2018) [5] proposed a multi-model neural network-based extractive summarizer. A multi-model architecture is used on is bidirectional RNN and the other is CNN. This method encodes the text and images using this architecture and the probability of the sentence is calculated using a logistic classifier that has text coverage and text redundancy as features. The dataset used the Daily mail corpus by gathering the images from the internet. Encoding of the images is done using bidirectional RNN.

Jorge V. Tohalino et.al (2018) [6] has proposed a multi-layered neural network model to generate the extractive summary based on the sentences having high importance in the context for multi-documents. A graph which consists of nodes and edges is built with sentences and words respectively. The datasets used are CST News for Portuguese multi- document summarization and DUC-2002, DUC-2004 for English multi-document summarization. The word embedding feature is lacking in this work for generating a better summary.

J.N.Madhuri et.al (2019) [7] has proposed a method to generate an extractive summary using sentence ranking technique using the term frequency after the removal of stop words. It works for any kind of text but cannot semantically distinguish sentences. The evaluation is done using the MSWord summarizer and human summarized summary. The summaries are converted to mp3 format so that it is easy to evaluate or know the summary.

Nikhil S. Shirwandkar et.al (2018) [8] has proposed an approach using both Restricted Boltzmann machine and Fuzzy logic to identify the important sentences. Two summaries for the same input document are obtained and then the final summary is obtained by considering both the summarization get meaningful and no loss of information. The author did summarization for only a single document. The f-measure obtained is 84%.

Mahmood Yousefi-Azar et.al (2016) [9] has proposed a text summarizer for a query-oriented system using an unsupervised deep learning network which is a deep auto-encoder to calculate feature space from term frequency. This is a stochastic machine. The experiments are done on SKE and BC3 email datasets. A semi-supervised learning approach works better because it can be used for unlabelled data.

Aditya Jain et.al (2017) [10] has proposed a neural network-based text summarization, the input of the nodes is the values of different features [11] and the dataset used is

DUC-2002. The summary is compared with four online summarizers. The accuracy of summarization can be exalted using a large training dataset for a neural network.

## II. PROPOSED METHOD

The author used statistical methods for generating precis. There are seven features which are chosen to precis the information. All the seven features that are calculated for each sentence and the sentences are given a ranking from which the threshold known as the compression ratio [6] is chosen which tells how many sentences should be present in the summary. Suppose 'k' sentences should be included in the summary then 'k' sentences with the highest rank are added to the summary. The sentences with a certain compression ratio should be in ascending order so that the context of the article will not change.

### A. Architecture

- 1) Input Document: To generate a summary some input should be given in the file format “.txt”. The text documents of sports articles are considered for summarization and are valid since the author has proposed a method for summarizing sports articles.
- 2) Pre-processing: Stop word removal and stemming are done to the text considered as input after lower casing. Fig2 shows various techniques of pre-processing the author used. In the lower casing, the whole document is converted into lowercase alphabetical letters. This is done to remove the ambiguity of the words with different casing. The stop words in the document are removed from the document because these are the most frequent words like articles, prepositions, conjunctions which don't add any value in defining the importance of a sentence. The stop words are removed using nltk (natural language tool kit) library. Stemming [4] is referred as getting the words reduced to their root form. It removes the different forms of the same word present in different sentences. For example, eating, ate and eat are identified as eat.

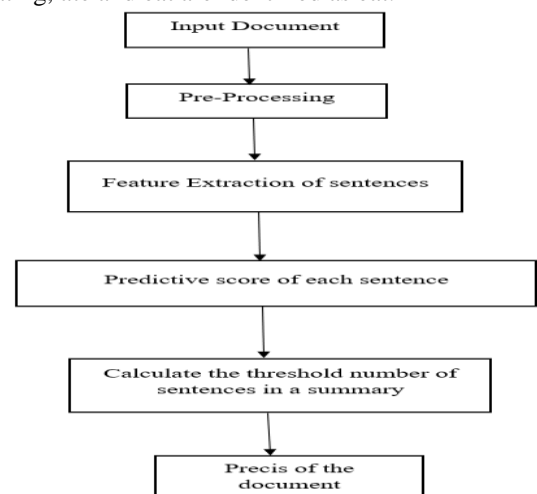


Fig.1 Architecture of the proposed method

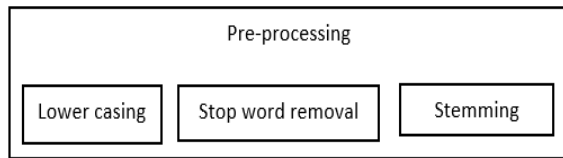


Fig.2 Steps included in Pre-processing

3) Feature Extraction – After pre-processing the text, sentence features are extracted to calculate the predictive scores.

- TF-ISF: Term frequency and inverse document frequency (TF-IDF) [4] are used for information retrieval systems. In this paper, the summarization is done for a single document of sports article and Term frequency inverse sentence frequency (TF-ISF) is calculated as follows.

$$TF - ISF(i, j) = TF(j) * \log\left(\frac{N}{ISF(j)}\right) \quad (1)$$

$$TF - ISF(i) = \sum_{j=0}^n \frac{(TF-ISF(i,j))}{n} \quad (2)$$

Where  $TF_j$  indicates the term frequency of a word in a sentence,  $ISF_j$  is the inverse sentence frequency of a  $j^{th}$  word in a sentence,  $TF-ISF_i(2)$  is the Term frequency and inverse sentence frequency of  $i^{th}$  sentence in a document,  $n$  cites to the number of words in a sentence,  $N$  cites to the number of sentences in a document.

- Sentence length: This feature provides the less weightage to short sentences as short sentences are relatively less important when compared to long sentences [2]. It is measured by calculating the ratio of no of words in the sentence to the no of words in the longest sentence of the document. This feature's value is calculated using eq. (3).

$$sent. length = \frac{no\ of\ words\ in\ sentence}{no\ of\ words\ in\ longest\ sentence} \quad (3)$$

- Sentence position: In a document the numerical value of the sentence's position is gauged(4). The position is evaluated as the normalized percentile score in the range of 0 to 1.

$$sent. position = \begin{cases} 0, & \text{if first sentence} \\ 1, & \text{if last sentence} \\ \frac{p}{n}, & \text{if } 0 < p < n \end{cases} \quad (4)$$

$p$  is the position of the current sentence and  $n$  is the total number of sentences.

- Cosine similarity: Analogy between sentences is calculated using this property. This briefs how much similar this sentence with other sentences in a document is [7]. A vector for each sentence is assessed and (5) is practiced obtaining the score of the feature.

$$cosine\ similarity = \frac{A \cdot B}{|A||B|} \quad (5)$$

$A$  is the first vector and  $B$  is the second vector with which it is compared

- Existence of proper noun: Proper names broaching to places and people might be useful to decide the relevance of a sentence. This binary feature, with value '1' if a sentence contains these proper names and 0 otherwise.
- Existence of pronoun: Pronouns cites to the persons which are described in the previous sentences. This binary

feature, with value '1' if a sentence contains these proper names and 0 otherwise.

- Sentence containing scores: Since in sports the score of the team or the individual participant is important the score is taken into consideration. This binary feature, with value '1' if a sentence contains these proper names and 0 otherwise.
- 4) Predictive score of each sentence: The predictive score is calculated by considering the mean of cosine similarity feature and TF-ISF feature and taking the sentence length and sentence position and addition of the remaining features and the score for each sentence is given known as the predictive score.
  - 5) Calculate the threshold number of sentences in a summary: The threshold value known as the compression ratio tells us how much percent of the total document is presented as a summary.
  - 6) Precis of the document: This is the output of the text document which is the summary. Suppose 'k' be the threshold and the 'k' sentences with the highest predictive score should be in ascending order so that the context of the article will not change.

## B. Algorithm

- Step1: Take text file as an input
- Step 2: Split the entire document into sentences.
- Step 3: Convert all the sentences into lowercase.
- Step 4: Split the sentences into words and remove the stop words from the obtained words.
- Step 5: Stemming for all the words after stop word removal is done using nltk toolkit.
- Step 6: For all the sentences
- Step 7: Calculate the TF-ISF for each word after stemming and take the mean of all the values of a sentence and one value is obtained which is the TF-ISF for one sentence.
- Step 8: Repeat step 7.
- Step 9: Calculate the sentence length for each sentence.
- Step 10: Check for proper noun in a sentence if present update to 1 else 0.
- Step 11: Check for pronoun in a sentence if present update to 1 else 0.
- Step 12: Calculate the sentence position using (4)
- Step 13: Check if a value is present in sentence if present update to 1 else 0.
- Step 14: Convert the sentences into vectors.
- Step 15: For all the sentence vector
- Step 16: For all the next sentence vector
- Step 17: Calculate the cosine similarity. Repeat 16
- Step 18: Repeat 15
- Step 19: Calculate the mean of all the cosine similarity with respect to other sentences and obtain a value for each sentence.
- Step 20: Calculate the mean of TF-ISF and cosine similarity values, obtain predictive score.
- Step 21: Sort the predictive scores and list the top sentences based on the compression ratio.
- Step 22: Add the sentences to the final summary.
- Step 23: Print the final summary.

C. Pseudo Code

- Pre-processing
 

```
f=open("input.txt",'r')
m= [] ([ ] : Empty list)
for line in f:
    m.append(line.split(' '))
sentence= []
for i in range(len(m)):
    sentence.append(m[i].lower())
words_after_stemming=[]
for i in range(len(sentence)):
    if(sentence[i] not a stop word):
        words_after_stemming.append(PorterStemmer().stem(sentence[i]))
```
- TF-ISF
 

```
tfisf=[]
for i in range(len(words_after_stemming)):
    for j in range(len(words_after_stemming[i])):
        tf=words_after_stemming[i].count(words_after_stemming[i][j])
        isf=math.log(N/c); tfisf[i]=tf*isf
```
- Sentence length
 

```
maximum_length=max(words_after_stemming.count())
sentence_length=[]
for i in range(len(words_after_stemming)):
    sentence_length.append(words_after_stemming.count()/maximum_length)
```
- Proper Noun
 

```
propernoun=[]
for i in range(len(words_after_stemming)):
    if(pos_tag(words_after_stemming[i].split())[0][1]=='NN'):
        propernoun.append(1)
    else:
        propernoun.append(0)
```
- Pronoun
 

```
pronoun=[]
for i in range(len(words_after_stemming)):
    if(pos_tag(words_after_stemming[i].split())[0][1]=='PRP'):
        pronoun.append(1)
    else:
        pronoun.append(0)
```
- Sentence position
 

```
sentenceposition=[]
for i in range(1,len(words_after_stemming)+1):
    sentenceposition[i-1]=(i/len(words_after_stemming))
```
- Sentence containing scores
 

```
score=[]
for i in range(len(words_after_stemming)):
    if(sords_after_stemming[i] in '0123456789'):
        score.append(1)
    else:
        score.append(0)
```
- Cosine similarity
 

```
cosine_similarity=[]
generate vectors for words
```

- apply (5)
- Mean of TF-ISF and cosine similarity
 

```
score=[]
for i in range(len(cosine_similarity)):
    score+=mean(cosine_similarity[i],tfisf[i])
```
- Sort predictive score
 

```
sort.score()
summary=[]
summary.append(score(i,compression_ratio))
```
- Print summary
 

```
print(summary)
```

III. RESULTS

The author considered BBC sports article data set since it is the worldwide famous for its news. It consists of 737 articles for five different types of sports namely athletics, cricket, football, rugby, tennis and 101, 124, 265, 147, 100 articles respectively. The model is tested for these articles and performed with a good amount of accuracy by generating the relevant context leaving the unnecessary sentences. The author also considered different sports articles from New York Times, CNN during the testing of model for accuracy.

A. Sample Input

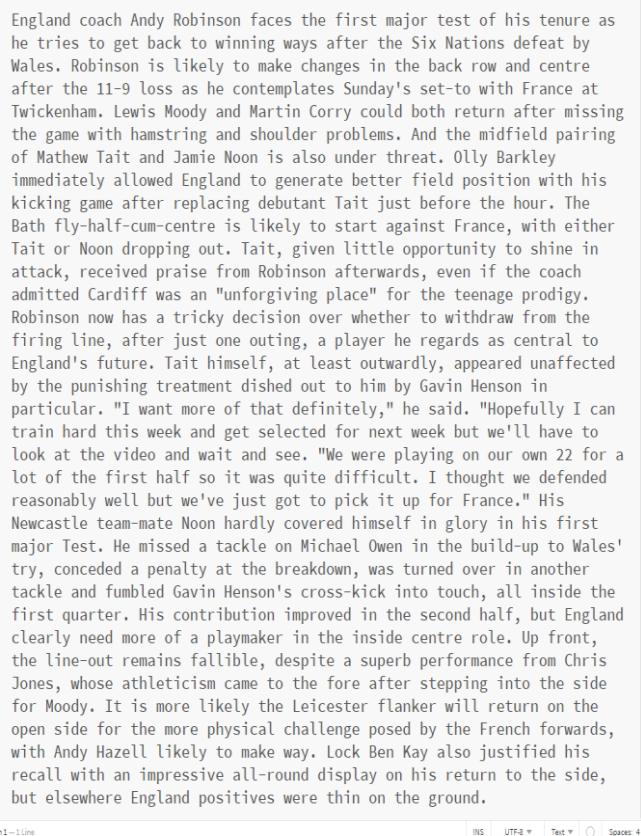


Fig.3 Sample input from dataset

B. Sample Output

[0, 1, 4, 6, 7, 10, 14, 16, 17, 18]

England coach Andy Robinson faces the first major test of his tenure as he tries to get back to winning ways after the Six Nations defeat by Wales. Robinson is likely to make changes in the back row and centre after the 11-9 loss as he contemplates Sunday's set-to with France at Twickenham. Ollie Barkley immediately allowed England to generate better field position with his kicking game after replacing debutant Tait just before the hour. Tait, given little opportunity to shine in attack, received praise from Robinson afterwards, even if the coach admitted Cardiff was an "unforgiving place" for the teenage prodigy. Robinson now has a tricky decision over whether to withdraw from the firing line, after just one outing, a player he regards as central to England's future. "Hopefully I can train hard this week and get selected for next week but we'll have to look at the video and wait and see. He missed a tackle on Michael Owen in the build-up to Wales' try, conceded a penalty at the breakdown, was turned over in another tackle and fumbled Gavin Henson's cross-kick into touch, all inside the first quarter. Up front, the line-out remains fallible, despite a superb performance from Chris Jones, whose athleticism came to the fore after stepping into the side for Moody. It is more likely the Leicester flanker will return on the open side for the more physical challenge posed by the French forwards, with Andy Hazell likely to make way. Lock Ben Kay also justified his recall with an impressive all-round display on his return to the side, but elsewhere England positives were thin on the ground.

Fig.4 Sample Output for compression ratio ( $\alpha=50\%$ )

IV. DISCUSSION

The evaluation measures used are for context evaluation which comes under co-selection. In co-selection there are three types of measures they are precision, recall, f-measure.

Precision (P) can be defined as the fraction of the number of sentences common in manual and automated summary to the number of sentences in the automated summary. Precision (P) can be procured using equation (6).

$$P = \frac{S(\text{manual}) \cap S(\text{auto})}{S(\text{auto})} \quad (6)$$

Recall (R) can be defined as the ratio of the number of sentences common in manual and automated summary to the manual summary. The recall can be calculated using equation (7).

$$R = \frac{S(\text{manual}) \cap S(\text{auto})}{S(\text{manual})} \quad (7)$$

F-measure (F) can be defined as the harmonic mean of precision and recall. F-measure (F) can be procured using equation (8).

$$F = \frac{2 * P * R}{P + R} \quad (8)$$

The System generated summary (SGS) is tested for accuracy with a manual summary and an online summarizer tool. The manual summary is given by one of the literates who have secured a degree in Literature. The online tool used for generating the summary is Text Compactor. The author has considered the compression ratio half of the original document.

Firstly, a manual summary is compared with SGS and a manual summary is compared with online summarizer tool for precision, recall, and f-measure. The documents tested are the sports articles from various newspapers including the New York Times, CNN. For analysing the accuracy, the author also checked it with BBC Sports Articles Dataset.

The Table-1 shows the number of sentences in each document taken from the news articles. The data of 10 documents is shown.

Title	Number of sentences in document	Number of sentences in SGS ( $\alpha=50\%$ )
Doc-1	11	5
Doc-2	16	7
Doc-3	15	7
Doc-4	8	4
Doc-5	36	19
Doc-6	15	7
Doc-7	39	20

Doc-8	27	14
Doc-9	19	10
Doc-10	32	16

The evaluation measures for some of the documents are given below.

Table.1 Shows number of sentences in document as well as SGS.

Title	Manual w.r.t SGS		
	P	R	F
Doc-1	0.8	0.8	0.8
Doc-2	0.7142	0.625	0.6662
Doc-3	0.7142	0.8333	0.7689
Doc-4	0.75	0.75	0.75
Doc-5	0.6842	0.7647	0.7219
Doc-6	0.7142	0.7142	0.7142
Doc-7	0.7	0.7368	0.7245
Doc-8	0.7142	0.7692	0.7401
Doc-9	0.8	0.889	0.8336
Doc-10	0.75	0.8	0.7741

Table.2 Comparison of performance between manual w.r.t SGS

Title	Manual w.r.t online summarizer tool		
	P	R	F
Doc-1	0.667	0.8	0.7274
Doc-2	0.6667	0.5	0.5714
Doc-3	0.5	0.667	0.5714
Doc-4	0.667	0.5	0.5714
Doc-5	0.7334	0.6470	0.6874
Doc-6	0.5714	0.5714	0.5714
Doc-7	0.834	0.834	0.834
Doc-8	0.5834	0.5384	0.56
Doc-9	0.667	0.667	0.667
Doc-10	0.5625	0.6	0.58

Table.3 Comparison of performance between manual w.r.t online summarizer tool.

	Average		
	P	R	F
Manual w.r.t SGS	0.7341	0.7681	0.7493
Manual w.r.t online summarizer tool	0.6452	0.6328	0.6341

Table. 4 The average of the Precision(P), recall(R), f-measure(F) for all the 10 documents

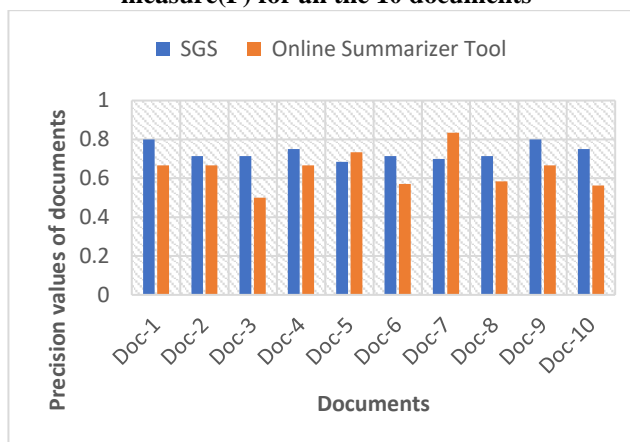


Fig.5 Comparison between SGS and Online summarizer tool

## V. CONCLUSION AND FUTURE SCOPE

The extractive text summarization has many tasks involved in generating the précis. The author proposed a statistical approach for generating an extractive précis of a sports articles that uses sentence ranking based on the selected features for the sentences. The most critical part is identifying the important sentences without loss of meaning from the given document. The manual summary is generated for all the documents and a summary from online summarizer tool and SGS is compared with them. Summing up using the proposed method on an average, 73% precision,

76% recall, and 74% f-measure is obtained which when compared with a manual generated summary and an online summarizer tool.

In nearly future, the proposed model can also be developed to a neural network model that takes the values of different features as input for generating an accurate précis by considering the contextual meaning. To increase the accuracy of the précis, various features can also be added.

## REFERENCES

1. Goularte, Fábio & Nassar, Silvia & Fileto, Renato & Saggion, Horacio. (2018). A Text Summarization Method based on Fuzzy Rules and applicable to Automated Assessment. Expert Systems with Applications. 115. 10.1016/j.eswa.2018.07.047.
2. Jo, Duke Taeho. (2017). K nearest neighbor for text summarization using feature similarity. 1-5. 10.1109/ICCCCEE.2017.7866705.
3. Anand, Deepa & Wagh, Rupali. (2019). Effective Deep Learning Approaches for Summarization of Legal Texts. Journal of King Saud University - Computer and Information Sciences. 10.1016/j.jksuci.2019.11.015.
4. P. Krishnaveni and S. R. Balasundaram, "Automatic text summarization by local scoring and ranking for improving coherence," 2017 International Conference on Computing Methodologies and Communication (ICCMC), Erode, 2017, pp. 59-64. doi: 10.1109/ICCMC.2017.8282539
5. J. Chen and H. Zhuge, "Extractive Text-Image Summarization Using Multi-Modal RNN," 2018 14th International Conference on Semantics, Knowledge and Grids (SKG), Guangzhou, China, 2018, pp. 245-248. doi: 10.1109/SKG.2018.00033
6. Valverde Tohalino, Jorge & Amancio, Diego. (2017). Extractive Multi-document Summarization Using Multilayer Networks. Physica A: Statistical Mechanics and its Applications. 503. 10.1016/j.physa.2018.03.013.
7. J. N. Madhuri and R. Ganesh Kumar, "Extractive Text Summarization Using Sentence Ranking," 2019 International Conference on Data Science and Communication (IconDSC), Bangalore, India, 2019, pp.1-3. doi: 10.1109/IconDSC.2019.8817040
8. N. S. Shirwandkar and S. Kulkarni, "Extractive Text Summarization Using Deep Learning," 2018 Fourth International Conference on Computing Communication Control and Automation (ICCUBEA), Pune, India, 2018, pp. 1-5. doi: 10.1109/ICCUBEA.2018.8697465
9. Azar, Mahmood & Hamey, Len. (2016). Text Summarization Using Unsupervised Deep Learning. Expert Systems with Applications. 68. 10.1016/j.eswa.2016.10.017.
10. Jain, Aditya & Bhatia, Divij & Thakur, Manish. (2017). Extractive Text Summarization Using Word Vector Embedding. 51-55. 10.1109/MLDS.2017.12.
11. K.Selvani Deepthi, Dr.Radhika Y(2015),"Extractive Text Summarization using Modified Weighing and Sentence Symmetric Feature Methods", in an International Journal of Modern Education and Computer Science, ISSN: 2075-0161 Volume 7 No-10 pp: 33-39, October 2015.

## AUTHORS PROFILE



**Sai Teja Polisetty** is currently pursuing final year in Computer Science and Engineering, Anil Neerukonda Institute of Technology and Sciences. His area of Interest is Machine Learning, Natural Language Processing, Big Data Analytics.



**K. Selvani Deepthi** is doctorate in computer science and engineering. She is working as Associate Professor, Department of CSE, Anil Neerukonda Institute of Technology. Her area of Interest is Machine Learning, Natural Language Processing, Text mining.



**Shaik Ameen** is currently pursuing final year in Computer Science and Engineering, Anil Neerukonda Institute of Technology and Sciences. His area of Interest is Machine Learning, Business Intelligence.



**Ravivarma G** is currently pursuing final year in Computer Science and Engineering, Anil Neerukonda Institute of Technology and Sciences. His area of Interest is Deep Learning, Natural Language Processing.



**Mounisha M** is currently pursuing final year in Computer Science and Engineering, Anil Neerukonda Institute of Technology and Sciences. Her area of Interest is Natural Language Processing.