# TEXT SUMMARIZATION

**Manju D*1, Radhamani V*2, Dhanush Kannan A*3, Kavya B*4, Sangavi S*5 Swetha Srinivasan*6**

*1*2 Assistant Professor, Department of Computing (Decision and Computing Sciences), Coimbatore Institute of Technology, Coimbatore, Tamil Nadu, India
*3*4*5*6 M. Sc (Integrated) Decision and Computing Sciences, Department of Computing, Coimbatore Institute of Technology, Coimbatore, Tamil Nadu, India

**1. Dr. Manju D**
Assistant Professor, Department of Computing (Decision and Computing Sciences), Coimbatore Institute of Technology, Coimbatore, Tamil Nadu, India (Corresponding Author)
Email ID: manju@cit.edu.in Mobile: +919865229679

**2. Radhamani V**
Assistant Professor, Department of Computing (Decision and Computing Sciences), Coimbatore Institute of Technology, Coimbatore, Tamil Nadu, India
Email ID: vradhamani@cit.edu.in Mobile: +919994314741

**3. Dhanush Kannan A**
M. Sc (Integrated) Decision and Computing Sciences, Department of Computing, Coimbatore Institute of Technology, Coimbatore, Tamil Nadu, India

**4. Kavya B**
M. Sc (Integrated) Decision and Computing Sciences, Department of Computing, Coimbatore Institute of Technology, Coimbatore, Tamil Nadu, India

**5. Sangavi S**
M. Sc (Integrated) Decision and Computing Sciences, Department of Computing, Coimbatore Institute of Technology, Coimbatore, Tamil Nadu, India

**6. Swetha Srinivasan**
M. Sc (Integrated) Decision and Computing Sciences, Department of Computing, Coimbatore Institute of Technology, Coimbatore, Tamil Nadu, India

## ABSTRACT

In the last few years, a huge amount of text data from different sources has been created every day. The enormous data which needs to be processed contains valuable detail which needs to be efficiently summarized so that it serves a purpose. It is very tedious to summarize and classify large amounts of documents when done manually. It becomes cumbersome to develop a summary taking every semantics into consideration. Therefore, automatic text summarization acts as a solution. Text summarization can help in understanding the huge corpus by providing a gist of the corpus enabling comprehension in a timely manner. This paper studies the development of a web application which summarizes the given input text using different models and its deployment.

**Keywords:** Text summarization, NLP, AWS, Text mining

# I. INTRODUCTION

Text summarization is an arduous technique of Natural Language Processing (NLP) given the problem in analyzing every word of the text in a document. An accurate analysis involving lexical knowledge of relations between words and semantic analysis is required which can be achieved with immense knowledge of words and the relations between them. Generating abstracts out of word documents in connection with similar words, various related content, etc. is a very exhausting and complex task. This is known as abstractive summarization. The type of summarization where the tool extracts the most significant and relevant contents and returns the text in a sequential and arranged manner is known as extractive summarization. Das and Martins (2007) have provided three major features for automatic text summarization.

- Summaries may be produced from a single or multiple document.
- Summaries should consist of important information.
- Summaries should be concise.

These features are vital, but an honest summary should even have other features such as coverage, non-redundancy, cohesion, relevancy, and readability (Shareghi and Hassanabadi, 2008; Sankar and Sobha, 2009; Parveen et al.,2016). Involving all the features in a summary is very challenging motivating us to enhance the composed summaries in all these aspects.

The summary created by any summarization system must contain the most sequential points of the original text document. When a computer program decreases the text to obtain such summaries, this technique is known as Automatic Text Summarization. In this type, the format is changed, but there will be no change in the semantics of the original corpus. The use of automatic summarization is increasing every day as the quantity of data has increased so much. There is a lot of interest in computer applications and technologies that can develop a precise summary by taking the variables such as length, indenting style and syntax.

AWS cloud is an advanced cloud platform and a very successful cloud service provider in a competitive situation. Various enterprises, individuals who are computer or application designers, even industries make use of cloud computing. It aids in making processes which are exhausting when done manually, easier. It also makes the best use of resources available thereby helping in constructing better relationships with the customer to take the business on various levels. AWS stands out from other cloud platforms in terms of flexibility and scalability.

# II. LITERATURE REVIEW

Vishal Gupta and Gurpreet Singh Lehal, “A Survey of Text Summarization Extractive techniques”. A clear picture of the idea of extractive summarization is painted by the author in this paper. The aspects used to develop the summary and a solution to solve the challenges has been clearly explained. According to the author "the importance of sentences is decided based on statistical and linguistic features of sentences"[2].

N.Moratanch and S.Chitrakala, "A Survey on Extractive Text Summarization". The author has emphasized on the importance of features like sentence level and word level features. They have categorized all extractive summarization methods into unsupervised and supervised methods and explained each method and have depicted few evaluation metrics[3].

Rajvardhan Oak, "Extractive Techniques for Automatic Document Summarization: A Survey". In this paper the author has elucidated different extractive summarization aspects. The author provides a comparative study of different extractive summarization methods and also explained two summarizer tools MEAD and summarist [4].

Selvani Deepthi Kavila and Dr.Radhika Y, " Extractive Text Summarization Using Modified Weighing and Sentence Symmetric Feature". In this paper, the author has emphasized on summarization of various fields' research papers. The author has overcome a few challenges like converting huge amounts of data to summaries and also removing unimportant sentences in the summary while using extractive methodologies by introducing a compression ratio that will help to find out the importance of each sentence [8].

Deepali K. Gaikwad and C. Namrata Mahender, " A Review Paper on Text Summarization". In this paper author has described both extractive summarization technique and abstractive summarization technique and have described text summarizers and summarization tools for Indian languages and have exhibited the comparison between performance of different methods [6].

Arpita Sahoo and Dr.Ajit Kumar Nayak, "Review Paper on Extractive Text Summarization". The authors of this paper have discussed various techniques and methods of text summarization and emphasized upon the challenges of extractive text summarization. [1]

Pradeepika Verma and Anshul Verma, "A Review on Text Summarization Techniques". The authors have explained the technicalities of document summarization. They have observed that many summarization techniques suffer from various challenges and hence effective summarization techniques are required. [7]

**Existing systems:**

Sentence Scoring based on Word Frequency:

The model initially assigns weight to each word of the given corpus. With the help of the weights assigned, the model will assign a score for each sentence. In the end, the model will rank the sentences according to the score and display a certain number of sentences as the summary. This model will not provide accurate results as it leverages the raw score of each sentence and provides the summary which can be greatly influenced by the length of the sentence.

TextRank using Universal Sentence Embeddings:

This model is derived from the popular PageRank proposed by Google Cofounders. A matrix is generated which contains the probability that a user will hop onto the next page. In the case of TextRank, the model generates a cosine similarity matrix which has the similarity of each sentence to each other. A graph is then generated from this cosine similarity matrix. The PageRank ranking algorithm is then applied to the graph to calculate scores for each sentence.

AYLIEN Text API:

AYLIEN Text API allows developers to extract meaning and insights from documents with ease. Specifically, its summarization endpoint returns a few key sentences, given an article as input. The Aylien Text API gives you the option to pass either a link to the article or its title and content as parameters. It lets us select a short mode or a default mode that further lets you control the number of sentences or percentage of sentences to be returned. The number of sentences the summary should have can be specified and the same number of sentences will be returned as the output.

# III. METHODOLOGY

## TEXT SUMMARIZATION:

Text summarization is the technique for creating a short and sharp summary of huge texts when targeting on the sections that convey important information, without losing the overall meaning.Summarization engines often aid in identifying the most crucial headings of the document.

Automatic text summarization aims to rework the long documents into shorter versions, which is tedious when done manually. Machine learning algorithms can be trained to understand the documents and find partitions that convey important verity and information before creating the required summarized texts. In scientific paper summarization, there is a certain amount of detail like cited papers and conference information which can be leveraged to identify dominant sentences in the original paper.

Summarization algorithms can either be abstractive or extractive based on the summary developed. Extractive algorithms form summaries by finding and pasting together relevant sections of the corpus. Therefore, they rely only on extraction of sentences from the corpus. This is why extractive methods yield naturally grammatical summaries and require relatively less linguistic analysis. In contrast, abstractive algorithms are almost human-like and mimic the process of paraphrasing a text, which may generate new text that is not present in the corpus. Condensed summaries and more human-like summarized texts are created using this technique. However, abstractive techniques are more difficult to execute than extractive summarization techniques. Current abstractive summarizers often depend on an extractive preprocessing component to produce the abstract of the document text is a worth mentioning one. According to Mani (1999)[5], a text summarization system filters the important information from the original corpus to generate a condensed version. Generally, the summarization process can be classified into three phases:

- Analysis of document text to obtain text representation.
- Transformation of text representation into summary representation.
- Transfiguration of summary representation into summary text to generate summary.

## MODELS USED:

### SUMY:

Sumy is a simple library and command-line utility for extracting summaries from HTML pages or plain texts. The package also contains a simple evaluation framework for text summaries.

Sumy is an open-sourced Python library to extract summaries from HTML pages and text files. The package contains an evaluation package for text summaries. Sumy offers many algorithms and methods for text summarization, some of them are:

- Luhn: heuristic method
- Latent Semantic Analysis (LSA)
- Edmundson heuristic method
- LexRank
- TextRank

**Using LexRank**

Unsupervised approach inspired by algorithm PageRank and HITS.LexRank is a graph-based approach for automatic text summarization. The scoring of sentences is done using the graph method. LexRank is used for calculating importance of sentences on the basis of the concept of eigenvector centrality in a graph representation of sentences. The main idea is that sentences "recommend" other similar sentences to the reader. Therefore, if one sentence is very similar to many others, it will likely be a sentence of great importance.

**SPACY**

spaCy is a free, open-source library for advanced Natural Language Processing (NLP) in Python. It is written in Cython and is designed to develop information extraction or natural language understanding systems.

spaCy is designed especially for production use and helps you build applications that process and "understand" large volumes of text. It can be used to build information extraction or natural language understanding systems, or to pre-process text for deep learning. It is built for production use and provides a concise and user-friendly API.

spaCy provides a variety of linguistic annotations to give you insights into a text's grammatical structure. This includes the type of words, like the parts of speech, and how the words are related to each other.

**GENSIM:**

Gensim is the fastest library for training vector embeddings in Python or any language. The core algorithms in Gensim use highly optimized & parallelized C functions. Gensim runs on all platforms including Linux, Windows and OS X, as well as any other platform that supports Python and NumPy. The summarize() function of the Gensim library returns a summarized version of the given text using a variation of the TextRank algorithm. The output summary will contain the most representative sentences and will be returned as a string, divided by newlines.

**NLTK:**

We developed a summarizer with the help of the NLTK library. We tokenized the corpus and removed the stopwords. We calculated the frequency of each word in the tokenized list and weighted every token according to their frequency thereby generating a summary.

## 3.1 FEATURES FOR EXTRACTIVE TEXT SUMMARIZATION:

The features of extractive text summarization identify and choose necessary sentences from the corpus and add them to the summary. Both word level and sentence level features are employed in text summarization literature. The features that can be applied to identify those sentences are:

## WORD LEVEL FEATURES

### 1. Content Word Feature

Keywords are important in finding the importance of the sentence. The sentence that consists of main keywords is most likely enclosed in the final summary. Keywords are the verbs, adverbs, nouns and adjectives that are commonly determined based on TF x IDF measure.

### 2. Title Word feature

The sentences within the original document that consists of words mentioned in the title have higher probabilities to contribute to the final summary since they function as indicators to the theme of the document.

### 3. Biased word feature

The sentences that contain biased words are mostly important. The biased words are a series of the predefined collection of words which may be domain specific. These are relatively important words which explain the theme of the document.

### 4. Upper case word feature

The words which are in uppercase such as "UNESCO" are referred to as necessary words and those sentences that contain these words are termed important in the context of sentence selection for the eventual summary.

### 5. Cue phrase feature

Cue phrases are phrases and words that provide the structure of the document flow and it is used as a feature in selection of sentences. The sentence that contains cue phrases (e.g. "summary", "because", "develop", "this information", "desire" etc.) are usually to be included in the final summary.

## SENTENCE LEVEL FEATURES

### 1 Sentence length feature

The sentence length plays a crucial role in identifying the key sentences. Shorter texts do not deliver important information and very long sentences also need not be included in the summary. The normalized length of the sentence is computed as the proportion of a number of words in the sentence to the
number of words in the longest sentence in the document.

### 2 Sentence location feature

The sentences that occur in the starting point and the conclusion part of the document are most likely to be important since most documents are sequentially structured with essential information in the beginning and the conclusion of the paragraphs.

**3 Paragraph location feature**

Similar to sentence location, paragraph location also plays a vital role in selecting the key sentences. A higher score is allocated to the paragraph in the peripheral segment(beginning and end paragraphs of the document).

**4 Similarity or cohesion feature**

Similarity features are vital to reduce the replications and arrange the sections in a sequential manner making it easy to comprehend. Similarity score is calculated between the sentences.

**Scraping text for summarization:**

We have also included a module wherein a web scraper is used to scrape the text off a given link or URL. The scraped text is later summarized using the above models. This module helps in saving time, since only the link has to be given. We have used Beautiful Soup and bs4 packages for web scraping.

**AWS Deployment:**

AWS provides various options for deploying your applications. Be it any application ranging from simple client-server or to complex workload involving applications, AWS offers customized deployment models according to the purpose. With the help of correct tools, AWS helps you deploy your application in a safe, secured and a rapid manner. It is the perfect platform to pick and choose servers and infrastructure that can handle the workload of our application. We used an EC2 instance to deploy our application with the help of Putty and WinSCP applications.

## IV. RESULTS AND DISCUSSION

Given an input text which generally takes ten minutes to read when read manually, our application manages to generate a summary of the document which takes only around a minute to read. Our default summarizer is Spacy summarizer.

There is another module in our application which helps to compare the different summaries generated by different summarizer models for the same input text. We evaluate the summaries based on the reading time for each summary and human interpretability which involves correct grammar and also should be understandable in natural language.

The input text given to our application takes about 3 minutes and 2 seconds to read. Table 1 compares the reading time of different models.

**Table 1.** Comparison of reading time of all 4 models

| S.No. | Model Type | Reading Time(minutes) |
|---|---|---|
| 1 | Spacy Summarizer | 0.94 |
| 2 | Gensim Summarizer | 0.82 |
| 3 | NLTK Summarizer | 0.855 |
| 4 | Sumy LexRank | 0.535 |

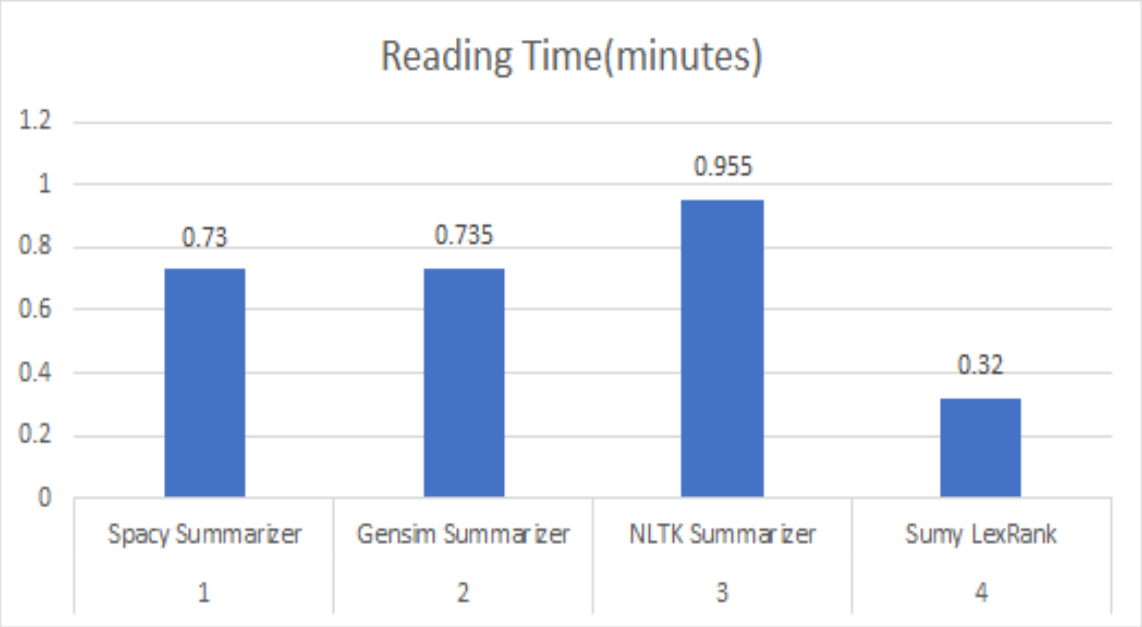


*Fig.1 Reading time (in minutes) of Summarizers*

**Table 2.** Comparison of word count of all 4 models

| S.No. | | Model Type | Word Count |
|---|---|---|---|
| 1 | | Spacy Summarizer | 89 |
| 2 | | Gensim Summarizer | 107 |
| 3 | | NLTK Summarizer | 131 |
| 4 | | Sumy LexRank | 53 |

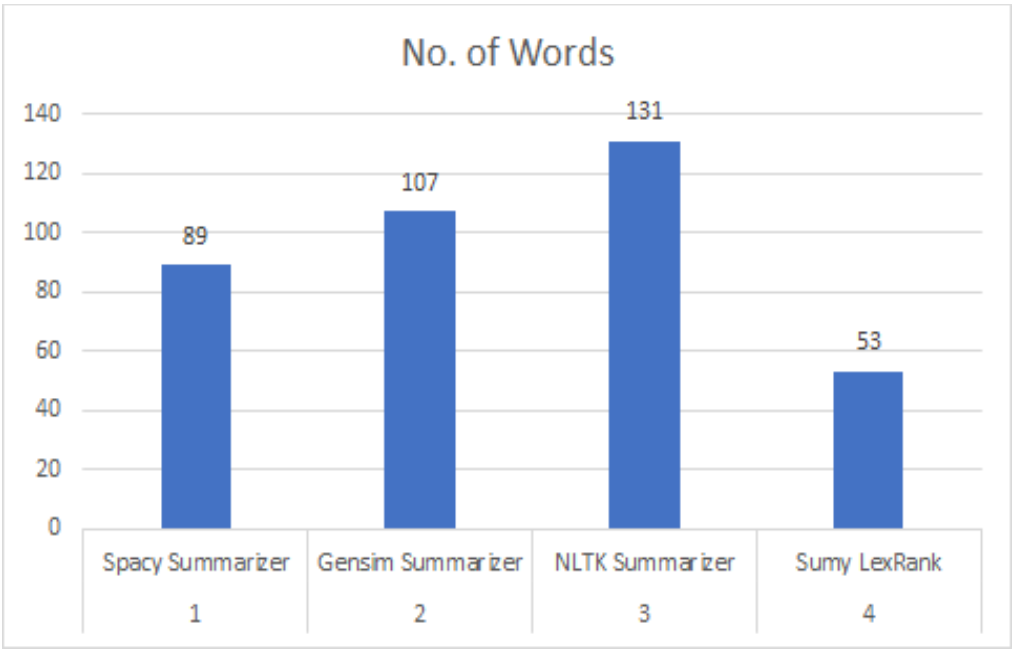


*Fig.1 Word Count of summarized text*

## V. LIMITATIONS

- Computers lack human intelligence and sometimes do not understand the language. This leads to a difficult and a very challenging task of automatic/ spontaneous summarization of text since a deeper analysis of document is necessary.
- It is very difficult to select a representative subset of sentences from the original corpus.
- It takes a lot of effort to generate a summary with minimum redundancy, maximum relevancy and referring elements of the document in the summary.
- Contradicting information cannot be included in the summary.

## VI. CONCLUSION

This paper studies the development of an application which helps to generate summaries of long text documents using 4 different models. It discusses the various features of extractive text summarization as well as the challenges faced. This paper talks about the deployment of applications in AWS cloud. It elucidates the features of the application developed.

## VII. FUTURE STUDY

The current application works only as a web application. The application could further be improved as a mobile application which helps in performing research very efficiently and in a time-saving manner. The application can include modules involving other natural language processing (NLP) functions like topic modelling, sentiment analysis, etc. We can include the abstractive summarization techniques for modules like sentence rephraser to help researchers write better research papers. We can develop our application as an API which is platform-independent.

.

## VIII. REFERENCES

[1] Arpita Sahoo and Dr.Ajit Kumar Nayak, “Review Paper on Extractive Text Summarization”, International Journal of Engineering Research in Computer Science and Engineering (IJERCSE) Vol 5, Issue 4, April 2018.

[2] Vishal Gupta and Gurpreet Singh Lehal, “A Survey of Text Summarization Extractive Techniques”, Journal of Emerging Technologies in Web Intelligence, Vol. 2, No. 3, August 2010.

[3] N.Moratanch and S.Chitrakala, "A Survey on Extractive Text Summarization", IEEE International Conference on Computer, Communication, and Signal Processing, 2017.

[4] Rajvardhan Oak, “Extractive Techniques for Automatic Document Summarization: A Survey”. International Journal of Innovative Research in Computer and Communication Engineering, Vol. 4, Issue 3, March 2016.

[5] Mani, I. (1999). Advances in automatic text summarization. MIT press.

[6] Saranya Mol C S, Sindhu L, “A Survey on Automatic Text Summarization”, International Journal of Computer Science and Information Technologies, 2014,Vol. 5 Issue 6.

[7] Pradeepika Verma and Anshul Verma, “A Review on Text Summarization Techniques”, Journal of Scientific Research, Volume 64, Issue 1, 2020

[8] Selvani Deepthi Kavila and Dr.Radhika Y, " Extractive Text Summarization Using Modified Weighing and Sentence Symmetric Feature", I.J. Modern Education and Computer Science, October 2015.

[9] R. Ferreira, L. de Souza Cabral, R. D. Lins, G. P. e Silva, F. Freitas, G. D. Cavalcanti, R. Lima, S. J. Simske, and L. Favaro. Assessing sentence scoring techniques for extractive text summarization. Expert Systems with Applications, 40(14):5755 – 5764, 2013

[10] Wen Xiao, Giuseppe Carenini, "Extractive Summarization of Long Documents by Combining Global and Local Context", 9th International Joint Conference on Natural Language Processing, pages 3011–3021, Hong Kong, China, November, 2019.