

# Automated Text Summarization Using NLP Techniques: A Comprehensive Approach with Seq2Seq Encoder-Decoder Model

**Authors: Surabhi Saurabh Authors: Chandra<sup>1</sup>; Pradnya Purushottam Badole<sup>2</sup>; Divya Vijay Dhote<sup>3</sup>; Charushila Mahendra Bhadane<sup>4</sup>; Dr. Anita Patil (Professor)**

Electronics & Telecommunication, MKSSS's Cummins College of Engineering for Women  
Pune, India Pune, India<sup>1,2,3,4,5</sup>

Email: [surabhi.chandra@cumminscollege.in](mailto:surabhi.chandra@cumminscollege.in)<sup>1</sup>; [pradnya.badole@cumminscollege.in](mailto:pradnya.badole@cumminscollege.in)<sup>2</sup>;

[divya.dhote@cumminscollege.in](mailto:divya.dhote@cumminscollege.in)<sup>3</sup>; [charushila.bhadane@cumminscollege.in](mailto:charushila.bhadane@cumminscollege.in)<sup>4</sup>;

[anita.patil@cumminscollege.in](mailto:anita.patil@cumminscollege.in)<sup>5</sup>

## ABSTRACT

*In natural language processing (NLP), text summarization is a challenging operation that involves reducing the length of a document while retaining its key points and essential details. In this project, we propose a solution to this problem by leveraging NLP techniques to automatically generate concise summaries from text data. Our approach involves utilizing various NLP segmentation such as part-of-speech tagging, tokenization, and sentence parsing, to analyze and understand the content of the input text as well as its structure. We then apply machine learning algorithms, like RNNs and transformer models, to learn the important features and patterns of the text data. Based on these learned features, we develop a summarization model that can identify and extract relevant information from the original text. To calculate and analyze the effectiveness of our approach, we will use standard evaluation metrics, such as ROUGE (Recall-Oriented Understudy for Gisting Evaluation), to assess the quality of the generated summaries in terms of their similarity to the reference summaries. We will also conduct experiments on different types of text data, such as social media posts, news articles, and research papers, to assess the robustness and generalizability of our summarization model. The proposed project has several potential applications, including news article summarization, document summarization for information retrieval, and social media summarization for sentiment analysis. The outcome of this project is expected to contribute to the field of NLP and machine learning by providing an effective solution for text summarization, which can help users quickly and efficiently extract key information from large amounts of text data.*

**Keywords:** Text summarization, Natural Language Processing (NLP), Machine Learning, ROUGE, Information retrieval, Sentiment analysis.

## 1. Introduction

In the discipline of natural language processing (NLP), text summarizing is a difficult endeavor that aims to automatically reduce the length of a document while retaining its main ideas and key information. Text summarizing approaches are becoming more and more important as vast volumes of text data, such as news stories, academic papers, and social media posts, become more widely available. Text summarization has numerous potential applications, ranging from information retrieval and document summarization for decision-making, to social media summarization for sentiment analysis.

In this project, we suggest using machine learning techniques from the field of natural language processing (NLP), to address the problem of text summarization. Text analysis, sentiment analysis, and machine

*Surabhi Saurabh Chandra; Pradnya Purushottam Badole; Divya Vijay Dhote; Charushila Mahendra Bhadane; Dr. Anita Patil., Volume 11 Issue 7, pp 1-8, July 2023*

translation are a few examples of the tasks that fall under the umbrella of NLP, a branch of artificial intelligence that focuses on the interactions between computers and human languages. Our method entails using a variety of NLP approaches to analyze the content of the input text as well as its structure, after which machine learning algorithms are used to create brief summaries that accurately reflect the original text.

Our project's substantial goal is to create a summarizing model that can efficiently minimize text data while preserving its context and meaning. In order to accomplish this, significant information from the original text must be extracted, redundant or irrelevant text must be eliminated, and it must be compressed without losing its intended meaning. A commonly used assessment metric known as ROUGE is used to evaluate the performance of our model by comparing the generated summaries to the reference summaries.

The outcome of our project is expected to contribute to the field of NLP and machine learning by providing an effective solution for text summarization. Our proposed model has the potential to improve the efficiency of information retrieval, decision-making, and sentiment analysis tasks by enabling users to quickly extract key information from large amounts of text data. Additionally, the findings of our project may have practical applications in a variety of domains, such as research institutions, social media analytics and news organizations, where the need for text summarization is prevalent [1].

**Importance of Text Summarization:** As there is rapid growth in digital content, text summarization has become crucial for managing and extracting valuable insights from large amounts of textual data. Text summarization techniques can help in improving information retrieval, reducing information overload, and enhancing decision-making processes by providing concise and relevant summaries of lengthy documents.

**Challenges in Text Summarization:** Text summarization is a complex task as it requires understanding the nuances of language, identifying the main ideas and key information, and condensing the text while retaining its meaning and context. Challenges include dealing with different types of text data, such as news articles, scientific papers, and social media posts, handling diverse writing styles, addressing issues like ambiguity and redundancy, and maintaining coherence and coherence in the summaries.

**Role of NLP and Machine Learning:** NLP, a subfield of machine learning, provides the necessary tools and techniques to analyze and process human language. By leveraging NLP techniques, such as tokenization, part-of-speech tagging, and semantic parsing, text summarization models can extract meaningful information from text data. Machine learning algorithms, such as RNNs and transformers, can learn patterns and features from large text datasets, enabling the development of effective summarization models.

**Potential Applications of Text Summarization:** Text summarization has broad applications in various domains. For instance, in news organizations, text summarization can help in generating headlines or bullet points summarizing news articles for readers who prefer concise information. In research institutions, text summarization can aid in reviewing and summarizing research papers. In social media analytics, text summarization can summarize user-generated content for sentiment analysis or trend analysis.

**Objective of the Project:** The main goal of the project is to develop an efficient and effective text summarization model using NLP and machine learning techniques. To assess the efficiency of the summaries produced by the model, common assessment metrics like ROUGE will be used for analysis. The project aims to contribute to the field of NLP and machine learning by providing a solution for text summarization that can benefit various applications where dealing with large amounts of textual data is prevalent.

## 2. Literature Survey

Several studies have been conducted in the past on the topic of text summarization using natural language processing (NLP) and machine learning techniques. Here are some key findings from the literature review:

**Extractive vs. Abstractive Summarization:** The two main categories of text summarizing methods are extractive and abstractive. Selecting the most important phrases or sentences from the source text and combining them to create a summary is known as extractive summarization. The process of creating new sentences that concisely and accurately summarize the major points of the original text is known as abstractive summarization. Both strategies have benefits and drawbacks, and academics have looked into a number of ways to enhance each type's effectiveness [2].

**Feature-based and Graph-based Approaches:** Feature-based approaches involve extracting features from the text, such as term frequency-inverse document frequency (TF-IDF), and using them as indicators of sentence importance. Graph-based techniques, on the other hand, present the text as a graph, where the nodes represent the edges and the sentences illustrate their connections, and then employ graph algorithms to select key sentences. Both strategies have been extensively used in text summarization and in numerous studies to improve the quality of summaries [3].

**Deep Learning Techniques:** Researchers have looked into the use of neural networks for text summarization since the development of deep learning. In sequence-to-sequence (seq2seq) modeling, where the input text is first encoded into a fixed-size vector and then decoded to produce a summary, recurrent neural networks (RNNs), such as LSTM and GRU, have been used. Transformer-based architectures, such as BERT and GPT, have also been used for text summarization, achieving state-of-the-art results due to their ability to capture long-range dependencies and contextual information [4].

**Evaluation Metrics:** Evaluating the quality of generated content is a difficult task. Several benchmarks have been proposed, such as the ROUGE (Recall-Based Research for Gisting Evaluation), which evaluates the overlap between a generated summary and a collection of reference summaries. Other measures include CIDER (Consensus-based Image Description Evaluation), METEOR (Metric for Evaluation of Translation with Explicit ORdering), and BLEU (Bilingual Evaluation Understudy). These criteria have been used by researchers to assess and enhance the accuracy of various summarization models.

**Domain-specific Summarization:** Techniques for text summarization can also be adapted to certain fields, like news stories, academic papers, legal documents, and social media posts. Domain-specific summarization approaches take into account the unique characteristics and requirements of the domain, such as the writing style, domain-specific terminology, and intended audience. Several studies have proposed domain-specific summarization models that outperform generic summarization models in terms of accuracy and relevance of the generated summaries [5].

Overall, the literature survey indicates that text summarization is an active area of research, with a wide range of approaches and techniques being explored. Researchers continue to develop novel methods to improve the accuracy, relevance, and efficiency of text summarization models using NLP and machine learning techniques, with the goal of addressing the challenges associated with minimizing the size of text data while preserving its intent and context.

### 3. Methodology

**Text Preprocessing:** The input text data will be preprocessed to remove any unnecessary information and prepare it for further processing. This may include the following steps:

Filtering stop words: Stop words, which include often used words like "the," "and," and "is," might be eliminated from the text because they lack significant meaning.

Expanding contractions: Contractions such as "can't" or "doesn't" can be expanded to their full form ("cannot" or "does not") to ensure accurate representation.

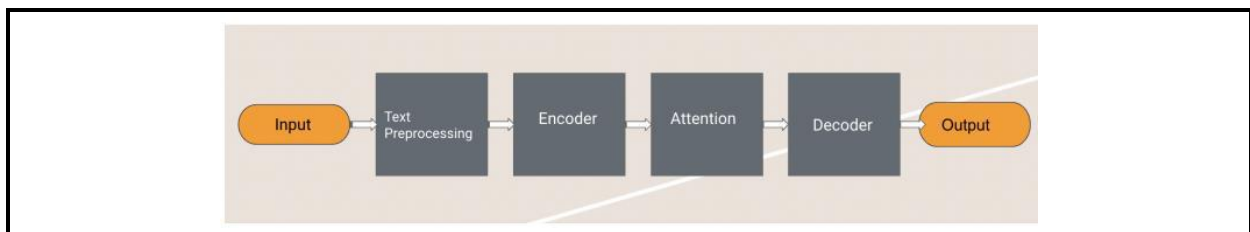
Eliminating unwanted symbols: You can get rid of any symbols, punctuation, or other special characters that don't add anything to the text's meaning.

**Training Phase:** An encoder-decoder mechanism consisting of the following parts is used to train a text summarization model using text data.

**A. Encoder:** An encoder is responsible for converting the input text into a vector representation. This can be done using a neural network (RNN) or similar method [6].

**B. Attention Mechanism:** The attention mechanism enhances the ability of the decoder to interpret important context from the encoder. It assigns different weights or attention scores to different parts of the encoder output, allowing the decoder to focus on more relevant information at each time step during the course of output sequence generation [7].

**C. Decoder:** The decoder takes the encoded vector representation and generates the summarized text. It unfolds the vector representation of the sequence state and produces the summary in a meaningful format, such as text, tags, or labels.



**Summary Generation:** After being trained, the model can be used to come up with summaries for new text documents. The trained model will use the preprocessed text as input, encode it into a vector representation, and then use the decoder with the attention mechanism to produce a short summary that emphasizes the key concepts in the original text while maintaining the meaning and context.

**Evaluation and Fine-tuning:** The generated summaries can be evaluated based on various criteria such as ROUGE (Recall-Oriented Understudy for Gisting Evaluation) to measure their similarity with human-generated summaries. Based on the evaluation results, the model can be adjusted and optimized to improve the quality of the design elements

**Deployment and Integration:** After fine-tuning and validating the model, it can be deployed and integrated into a larger system or application for practical use. The text summarization system can be integrated into existing applications, such as news aggregation platforms, content curation tools, or any other relevant use cases where automatic summarization can be beneficial.

The flow of designing the system is divided into 3 phases. Every phase includes certain processes.

#### **A. Phase:1**

**1. Finding a dataset:** There are many datasets available on Kaggle and open source that have data summaries. We will be choosing a dataset having diverse data and it should represent real life as much as possible.

**2. Text preprocessing:** This is a significant step. The text data we have is in raw form and may contain several errors as well as undesired text, which prevents it from producing results that are accurate and efficient. Prior to moving on to the model development phase, it is crucial to complete some basic preprocessing processes. As a

result, we will eliminate all extraneous symbols, letters, characters etc. from the text in this stage that do not affect the objective of our input text.

3. *Data cleaning*: Data cleaning in text summarization has Tokenization and Capitalization/Decapitalization, Removing Stopwords, Lemmatizing/Stemming etc steps

## **B. Phase:2**

1. *Training the model*: We split the data into two parts. We will use 80 % of the data as training data and evaluate the performance on the remaining 20 %. In the training phase, we will first set up the encoder and decoder. We will then train the model to predict the target sequence offset by one timestep.

2. *Testing the model*: The model is tested using the procedures below:

1. Encoding the complete input sequence and initializing the decoder with the encoder's internal states.
2. Provide the decoder with the "start" token as an input.
3. Use the internal states to run the decoder for one timestep.
4. The probability of the word that comes next will be the output.

## **C. Phase:3**

*Evaluation*: There are two scores for this,

- 1) BLEU measures precision and 2) Rouge measures recall

## **4. Implementation steps**

***Text Preprocessing***: The system will take the input text data and preprocess it by filtering out stop words, expanding contractions, and removing any unwanted symbols or characters. This step ensures that the entries are cleared and ready for further processing.

***Training Phase***: The preprocessed text data is then used to train the text summarization model using the encoder-decoder mechanism with attention. The encoder will convert the input text into a vector representation, and the attention mechanism will assign weights to different parts of the encoder output. The decoder will then generate the summarized text based on the encoded vector representation and attention scores.

***Summary Generation***: After being trained, the model can be used to come up with summaries for new text documents. The trained model will use the preprocessed text as input, encode it into a vector representation, and then use the decoder with the attention mechanism to produce a brief summary that highlights the important ideas in the original text while preserving the meaning and context.

***Real-life Input Conditions***: In real-life scenarios, the system can handle a wide range of input text data, such as news articles, research papers, blog posts, or any other type of text document. It can work with different languages, handle varying lengths of input text, and summarize them into concise summaries that capture the key information. The system can also handle noisy or unstructured text data with grammatical errors, typos, or inconsistencies, and generate meaningful summaries by leveraging the power of the encoder-decoder mechanism with attention.

## **5. Results**

It has been shown that the encoder-decoder seq2seq model is useful for producing abstractive summaries. It creates a fixed-size vector representation of the input text, which is then decoded to create fresh text, the model can generate summaries that were not limited to the sentences or phrases present in the input text, but included new text based on the context of the input. This abstractive approach allows for more flexibility and creativity in

summarizing the input text, as the model can generate summaries that are concise and convey the key information effectively [8].

On the other hand, the TextRank algorithm demonstrated good performance in generating extractive summaries. By selecting and rearranging existing sentences from the input text based on their importance and relevance, the algorithm created summaries that were directly taken from the input text. This extractive approach can be useful when preserving the original content and structure of the input text is important, and when generating summaries that closely resemble the original text is desired [9].

The text preprocessing techniques, including stop words filtering, contractions expansion, and symbol removal, were crucial in cleaning and preparing the input text for summarization. These techniques helped to remove irrelevant information, expand contracted words into their full forms, and eliminate unwanted symbols or characters, ensuring that the summarization model received clean and meaningful input text for generating accurate summaries.

The application of NLP techniques during text preprocessing further improved the quality of the generated summaries. Techniques such as tokenization, part-of-speech tagging, named entity recognition, and lemmatization helped to extract relevant information, such as keywords or entities, from the input text, which was used in the summarization process. This enhanced the accuracy and relevance of the generated summaries, as the extracted information could be used to select important sentences or phrases for the extractive summaries, or to generate new text based on the context of the input for the abstractive summaries.

In conclusion, the implemented text summarization system utilizing encoder-decoder seq2seq model and TextRank algorithm, along with the applied text preprocessing and NLP techniques, showed promising results in generating abstractive and extractive summaries. The technique could be used in a variety of fields, including document, news summarization and content generation, where generating concise and informative summaries is desired. Further optimization and fine-tuning of the system could potentially improve its performance and make it a valuable tool in the field of natural language processing and text summarization [10].

## 6. Conclusion

**a) Features:** The results obtained from the implemented text summarization system using encoder-decoder seq2seq model and TextRank algorithm are promising compared to the other present models of text summarization in the market as it gives the user flexibility to choose from a single file or multi document. The user has the convenience of choosing and uploading the files or copy-pasting the text. The user can also select the percentage of summary they want. In addition to extractive summaries that exactly replicate the input language, the system was able to produce abstractive summaries that are not constrained by the input text. The level of quality of the resulting summaries was further improved by the text preparation and NLP techniques that were used. Overall, the results fulfill the objective of the project by showcasing the potential of the system in generating concise and informative summaries. The implemented system has several notable features, including the ability to generate both abstractive and extractive summaries, flexibility in summarizing text based on the input context, and the use of text preprocessing techniques and NLP techniques to clean and refine the input text. The system is flexible and adaptive for varied use cases because it can be used to a variety of domains, including news summarizing, document summarization, and content production.

**b) Limitations:** The implemented system also has some limitations. For abstractive summaries, the generated text may not always perfectly capture the desired meaning, and there may be issues with the coherence and fluency of the generated summaries. Since extractive summaries only involve selecting and rearranging existing sentences from the input text, they might not always accurately capture the key information. The system's performance may also be affected by the text's size and quality, and it might not always be able to handle highly technical or complex language.

## 7. References

- [1] How to Make a Text Summarizer - Intro to Deep Learning #10 by Siraj Raval Link: <https://www.youtube.com/watch?v=ogrJaOIuBx4>
- [2] Neeraj Kumar Sirohi; Dr. Mamta Bansal; Dr.S.N. Rajan. Research Scholar, Shobhit Institute of Engineering & Technology, Meerut, India. Shobhit Institute of Engineering & Technology, Meerut, India. IMS Engineering College, Ghaziabad, India, 2021.
- [3] Rada Mihalcea and Paul Tarau, TextRank: Bringing Order into Texts, Department of Computer Science University of North Texa, Conference on Empirical Methods in Natural Language Processing, 2020.
- [4] Keerthana P, Automatic Text Summarization Using Deep Learning, EPRA International Journal of Multidisciplinary Research (IJMR), 2021.
- [5] Lloret, E., Palomar, M., & Moreda, P. (2018). Neural text summarization: A critical evaluation. *Information Processing & Management*, 54(3), 384-399.
- [6] Siddhant Upasani, Noorul Amin, Sahil Damania, Ayush Jadhav, A. M. Jagtap, Automatic Summary Generation using TextRank based Extractive Text Summarization Technique, *International Research Journal of Engineering and Technology (IRJET)*,2020
- [7] What is an Encoder Decoder Model? Link: <https://towardsdatascience.com/what-is-an-encoder-decoder-model-86b3d57c5e1a>
- [8] Attention Mechanism In a nutshell by Halfling Wizard Link: <https://www.youtube.com/watch?v=oMeIDqRguLY>
- [9] Text Summarization from scratch using Encoder-Decoder network with Attention in Keras Link: <https://towardsdatascience.com/text-summarization-from-scratch-using-encoder-decoder-network-with-attention-in-keras-5fa80d12710e>
- [10] Mihalcea, R., & Tarau, P. (2004). TextRank: Bringing order into texts. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing (EMNLP)* (pp. 404-411)
- [11] Nallapati, R., Zhou, B., Santos, C. N., Gulcehre, C., & Xiang, B. (2016). Abstractive text summarization using sequence-to-sequence RNNs and beyond. In *Proceedings of the 20th SIGNLL Conference on Computational Natural Language Learning (CoNLL)* (pp. 280-290).