JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue
JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)
An International Scholarly Open Access, Peer-reviewed, Refereed Journal

Designing a system using NLP for summarizing a text and retaining crucial points in a text

Prof. Kalyani Pendke[1], Achal Lute[2], Tanuja Kadwe[3], Bhavik Balpande[4], Chitrani Somkuwar[5], Utkarsh Zode[5]

[1,2,3,4,5,6] Department of Computer Science and Engineering,

Rajiv Gandhi College of Engineering and Research, Wanadongri, Nagpur, Maharashtra, India.

Abstract: Text summarization is a powerful solution to the overwhelming amount of textual data available to us. By using NLP techniques, algorithms can analyze and distill large amounts of text into concise summaries. Extractive summarization involves selecting relevant sentences from the original text, while abstractive summarization generates a summary that captures the essence of the information. While abstractive summarization can be more challenging, it often produces more comprehensive and readable summaries. As NLP techniques continue to evolve, we can expect more accurate text summarization solutions to emerge. Our tool offers a comprehensive view of the abstractive technique and allows users to perform summarization using pre-trained models. We have used the Text to text transformer (t5) base model which is a Natural Language Processing (NLP) Model implemented in the Transformer library, using the Python programming language. We have also implemented a summary range feature and the ability to copy the summarized text for future use.

Keywords: Natural Language Processing (NLP), text summarization, abstractive summarization.

1. INTRODUCTION

Natural Language Processing (NLP), a pivotal field within artificial intelligence, plays a crucial role in advancing emerging technologies. By enabling computers to comprehend, interpret, and manipulate human language, NLP bridges the gap between communication and machine understanding. NLP encompasses diverse disciplines such as computer science and computational linguistics and encompasses applications like sentiment analysis, speech recognition, text classification, machine translation, question answering, and more. This multidisciplinary approach empowers computers to interact with and understand human language, unlocking new frontiers of technological innovation and expanding the boundaries of human-computer interaction

A summary is a concise representation of key information extracted from one or more source texts. It distills the essence of the original text into a brief form, capturing the essential meaning and conveying it in a concise manner. Automatic text summarization employs advanced techniques to condense the source text while preserving its semantic meaning. By leveraging sophisticated algorithms and natural language processing, automatic text summarization aims to present a shorter version of the source text that retains its essence, facilitating efficient information consumption and enhancing communication in a concise and meaningful manner [10].

Automatic text summarization is designed to extract relevant and key points from vast amounts of data. With the ever-increasing volume of information available on the internet, collecting essential data becomes challenging and time-consuming. The goal of automatic text summarization is to efficiently and effectively condense large data sets into concise summaries, enabling quicker and more efficient information extraction for improved decision-making and knowledge acquisition. The use of automatic text summarization makes it easier for users to collect important data from huge information [11]. A summary is thus helpful as it saves time and recovers a massive document's data. Before this time, it was done by manual labor, but automation has brought forth many advantages [12].

2. LITERATURE REVIEW ON TEXT SUMMARIZATION

The author Shrivarsheni employed various methods for text summarization including traditional extractive, abstractive, and advanced generative methods. To accomplish this, the author utilized Python libraries such as "genism" and "sumy" that offer unique algorithms for extractive summarization. For instance, the LexRank approach prioritizes higher-ranked sentences based on recommendations from similar sentences. Latent Semantic Analysis (LSA) extracts semantically significant sentences using singular value decomposition (SVD), and the Luhn Summarization algorithm is effective for handling low-frequency and highly frequent words (stopwords). These diverse approaches enhance the efficiency and effectiveness of summarizing large amounts of data for concise and meaningful summaries [1].

The author Madhav's article focuses on providing an overview of various Text Summarization approaches, with a specific emphasis on analyzing extractive approaches. The goal was to compare the results of different approaches that were experimented with, including Sentence Scoring based on Word Frequency. This approach assigned weights to words based on their frequency in the passage, which was the simplest of the three approaches explored. Another approach evaluated in the article was TextRank using Universal Sentence Encoder,

which aimed to generate summaries by leveraging universal sentence embeddings and the TextRank algorithm. The article aimed to provide unique insights and analysis on the effectiveness of these approaches in generating concise and meaningful summaries from text data [2].

Divakar's research analyzed the performance of five distinct text summarization algorithms, including TF-IDF, LexRank, TextRank, BertSum, and PEGASUS, using the Reddit-TIFU and MultiNews datasets. The evaluation was conducted using the Recall-Oriented Understudy for Gisting Evaluation (ROUGE) to obtain meaningful and impactful summary results. The research provided valuable insights and findings on the effectiveness of these algorithms in text summarization and contributed to the existing body of knowledge in this field. The study's balanced execution of algorithms and evaluation against state-of-the-art literature highlights its significance and relevance in the field of text summarization [3].

The author Elena focused on multilingual text summarization and proposed three approaches for generating summaries in English, Spanish, German, and French. These approaches included language-independent techniques, language-specific resources, and incorporating machine translation resources into a mono-lingual summarizer. The goal of the study was to identify the most effective approach for addressing this challenge. The JRC corpus was used for evaluation, and it was found that the approach which uses language-specific resources outperforms state-of-the-art multi-lingual summarizers. This research provides valuable insights for further advancements in text summarization research, particularly in the area of multilingual summarization [4].

The author Subash Voleti proposed one innovative approach to text summarization that involves using a data-efficient content selector that identifies key phrases in the source document. This bottom-up attention step enhances text compression and summary generation, surpassing other content selection models. The two-step process demonstrates significant improvements in ROUGE evaluation for widely used corpora such as CNN-DM and NYT. Notably, the content selector can be trained effectively with as few as 1,000 sentences, making it adaptable to diverse domains with minimal training data [5].

The author Frederic Kirstein suggested that Text summarization, the process of the compressing source text while preserving its information and meaning, poses a challenge for manual summarization of large documents. However, with the advancements in machine learning, text summarization is becoming an increasingly intriguing field of research. As ongoing research continues to progress, we can anticipate further advancements that will enable fluent and accurate summarization of lengthy text documents using NLP techniques. With these developments, we can confidently state that we have efficiently completed text summarization using NLP, as per the problem statement [6].

The author Sebastian developed a system using Neural network-based methods and the Baseline Approach for abstractive summarization tends to produce fluent outputs but struggles with content selection. However, a blended bottom-up summarization system has shown significant improvement in ROUGE scores by over two points on CNN-DM and NYT corpora. This indicates that solving this problem requires fine-tuned inference restrictions and a balanced end-to-end trained approach. Furthermore, the data efficiency of this technique allows for easy transfer to new domains with limited data points. Similar bottom-up approaches are being explored in other domains like grammar correction and data-to-text generation [7].

Ganest and Lapalme proposed an innovative approach for text summarization based on representing documents as categories and aspects. Information extraction rules are applied to select the best candidates for each aspect, and sentence generation patterns are used to create concise and well-written summaries from clusters of news articles on the same event. Abstraction schemes are designed for each theme or subcategory, generating rules based on verbs, nouns, and syntactic positions. The content selection module selects the best candidate for each aspect, leading to higher information density summaries compared to state-of-the-art methods [8].

The Author Syed, in his review of summarization processes, delves into the various methods employed by these systems and evaluates their effectiveness and limitations. One approach involves assigning scores to sentences in the source text and selecting the highest-scoring ones for the summary. These scores are generated based on features extracted from the sentence, with the mappings between features and scores as well as the coefficients in the linear combination all derived from a training corpus. This system is particularly useful for users with limited reading ability or background knowledge, as it simplifies complex concepts and draws on external sources to provide necessary context. By streamlining the summary generation process, these systems offer a powerful tool for making information more accessible and digestible [9].

3. LITERATURE REVIEW ON TEXT SUMMARIZATION

References	Objectives	Data set	Techniques used
1	Text summarization approaches for NLP	CNN-dailymail datasets	Lexrank, LSA, Luhn algorithms
2	Comparing text summarization techniques	STS benchmark dataset and companion dataset	Sentence Scoring based on Word Frequency, TextRank using Universal Sentence Encoder
3	Qualitative Analysis of Text Summarization Techniques and Its Applications in the Health Domain	Reddit-TIFU and MultiNews dataset.	Term frequency-inverse document frequency (TF-IDF), LexRank, TextRank, BertSum, and PEGASUS
4	Finding the Best Approach	CNN-dailymail datasets	The proposed approaches rely on i) language-

	for Multi-lingual Text Summarization: A Comparative Analysis		independent techniques; ii)language-specific resources; and iii) machine translation resources applied to a mono-lingual summarizer.
5	Text summarization using natural language processing and Google text-to-speech API	News, Blogs, and Articles datasets	Text Rank Algorithm. Text to Speech API, Unsupervised TextRank Algorithm, Advanced NLTK Techniques
6	Automatic text summarization in NLP	Data available to us on the Internet and in different archives	-
7	Bottom-Up Abstractive Summarization	Evaluate our approach on the CNN-DM corpus, and the NYT corpus, which are both standard corpora for news summarization.	Content selection problem as a word-level extractive summarization task, Bottom-Up Copy Attention, End-to-End Alternatives
8	A Study on Abstractive Summarization Techniques in Indian Languages	Source document consisting of various languages	Structured-based approach and Semantic-based approach
9	Text Summarization using Natural Language Processing	WordNet dataset	TF-IDF and Countvectorizer

4. PROPOSED APPROACH

In this paper, an abstractive summarization approach is employed to solve the problem related to extractive summarization, semantic representations, document summarization, length limit of summary generated text, and scalable & easy environment for accessing summarized data& storing data.

Abstractive text summarization is a complex and sophisticated process that involves using advanced natural language processing techniques to comprehend the meaning and context of the original text, and then creating a new summary that captures the essence of the source material. Unlike extractive summarization, which simply selects and combines key phrases from the text, abstractive summarization requires the system to generate entirely new phrases and sentences, often requiring a deep understanding of the subject matter and creative use of language. This approach can result in more accurate and informative summaries, but it also presents significant technical challenges in terms of language modeling and content generation.

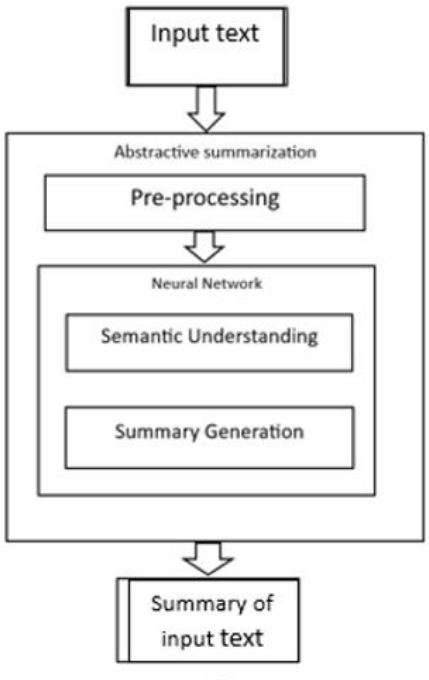

Figure 1: System architecture.

The general flow of architecture is started by taking input from the user then that input text undergoes a number of abstractive summarization steps in which we first do pre-processing of data in this step the redundant data is removed then the output of pre-processing step is undergoes into a neural network.
A network where with the help of semantic understanding summary will be generated and we get the output as the summary of input data.

4.1 Modules

Module I: Input
In this module, we have developed a powerful text summarization solution that can handle various input sources, including plain text from any source and text files in various formats.
Module II: Processing
In this module, input text undergoes various processing steps with respect to different algorithms like-

a. Data Pre-processing: In data pre-processing first the load_data() function reads the dataset and returns a pd.Data Frame object containing the raw data and cleaning, splitting, and tokenizing our input data.
b. Model Definition: In model definition, the encoding and decoding of data is done with the help of a T5-base encoder and decoder
c. Training and Evaluation: Here the model training is done using the fit () method.
d. Inference: Here with the help of different functions text summarization and evaluation is done.

Module III: Application Front-end development
In this module we develop the application front end as we connect the user with our application algorithm so that they can easily access our system it is the interface between the user and the system.

Module IV: Output
In this Module, after module II is done we get the summarized output data, after clicking on submit button of our application the summarized output is visible to the user.

5. DATASET

The dataset that we used is the **"news_summary"** dataset which was available on the Kaggle website [13].
The **"news_summary"** dataset can be used for several NLP applications, including sentiment analysis, text categorization, and text summarising. For researchers and programmers who are interested in creating machine-learning models for text analysis, it is a helpful resource.
The dataset has 4515 samples and includes the following information: Author_name, Headlines, Article URL, Short text, and Complete Article. I only scraped the news articles from the Hindu, Indian Times, and Guardian while gathering the condensed news from Inshorts. From February to August of 2017 are covered.

We are interested in the "text" column which consists of the Short text or summary of the news and the "text" column which consists of the Complete Article or original news.

6. ALGORITHM/DESIGN

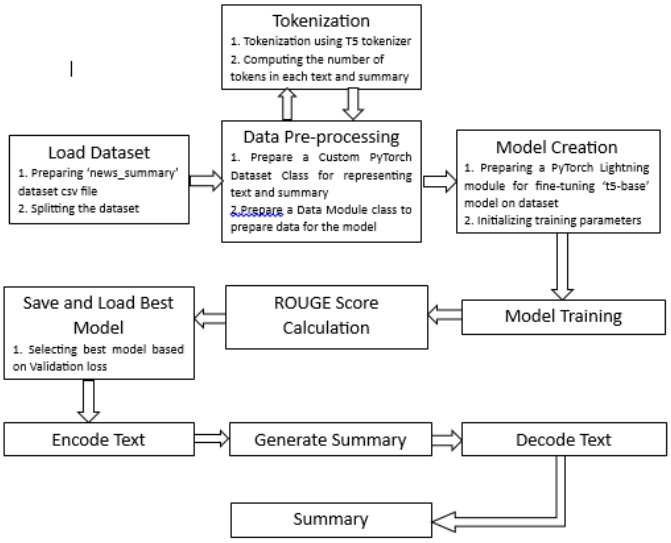

Figure 2: A basic step for implementation.

Firstly, the required libraries and modules are imported, including PyTorch, PyTorch Lightning, and Transformers. The dataset is loaded and pre-processed using the **news_summary** Dataset class. The dataset is split into training, validation, and testing sets, and is tokenized using the t5-base Tokenizer from the transformers library.

The **news_summary** Data Module class is created to organize the data and the data loading process. The class is responsible for downloading and pre-processing the dataset, creating data loaders for the training, validation, and testing sets, and defining the size of the vocabulary.

The Summary Model class is defined as a PyTorch Lightning module, which is responsible for training the model. The class contains a t5-base encoder and a decoder, both from the transformers library. The t5-base encoder is used to encode the input text, and the decoder is used to decode the summary. The model is trained using the Lightning Module API from PyTorch Lightning.

The Model Checkpoint callback and the Tensor Board Logger are defined to track the model's performance during training and save the best-performing model. The Trainer is created with the defined callbacks and logger. The trainer is responsible for training the model using the fit method, which takes in the SummaryModel instance and the **news_summary** Data Module instance. After the training is complete, the best-performing model is loaded from the saved checkpoint, and the generate_summary function is defined to generate summaries using the best model. The function takes in the input text and uses the best model to generate a summary.

The summarize function is then defined, which takes in the input text, encodes it using the encode_text function, generates a summary using the generate_summary function, and decodes the summary using the decode_summary function.

7. RESULT

1) Plain text summarization

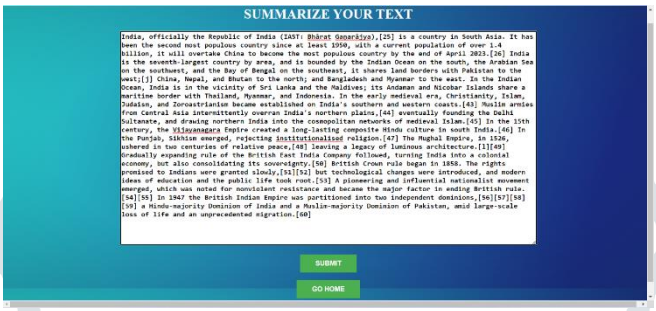

figure 3: Input page of Plain text summarization.

Here we give the input text (we take some information about India from Wikipedia) and after clicking on submit button we get a summary of the input plain text as output.

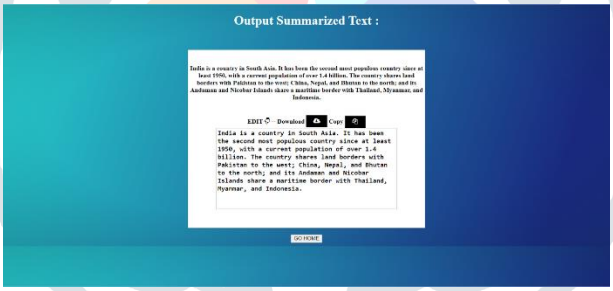

figure 4: Output page of Plain text.

2) Document summarization

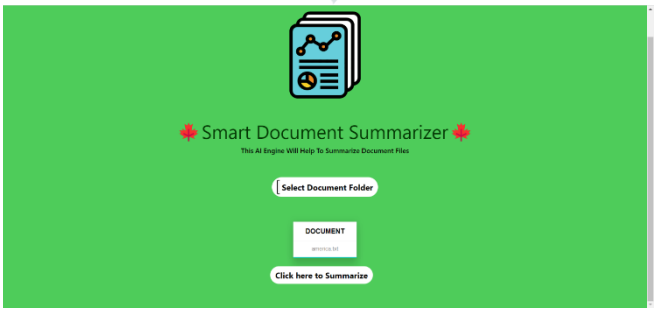

Figure 5: Input document summarization page.

Here we get an option like “select document folder” from which we select the America.txt file as the input document consisting of short paragraph information of America from Wikipedia and after clicking here to the summary button we get a summary of our input file as an output.

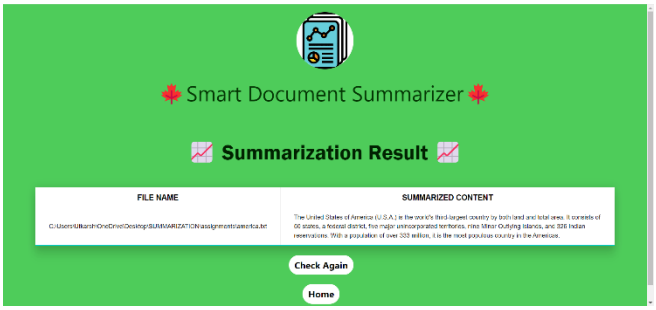

Figure 6: Output page of document summarization.

8. CONCLUSION AND FUTURE SCOPE

One of the best methods for natural language processing is automatic text summarization. After reading and analyzing numerous texts on text summarization, we have come to the conclusion that there are two methods for doing so Extractive summarization and Abstract summarization method. Abstractive summarization takes into account the full text and develops a summary based on its primary ideas. Because it takes into account all the main ideas, this type of summarization is more accurate than extractive ones. Text summarization is an intriguing area of machine learning that is attracting more attention. We successfully finished text summarization using NLP in accordance with the task definition. Our study led us to the conclusion that the carefully crafted T5 base model produced excellent results and produced a sound and fluid summary for a given text material than the other models where the other model does not capture the full meaning or context of the original text.

The future of abstractive text summarization lies in improving the accuracy of the systems, enabling multilingual and domain-specific summarization, developing interactive summarization systems, and integrating it with other applications to provide personalized and accurate summaries that can transform the way we process and consume information.

9. REFERENCES

[1] Shrivarsheni, “Text summarization approaches for NLP”, October 24, 2020.

[2] Madhav Thaker, “Comparing text summarization techniques”, Mar 25, 2019.

[3] Divakar Yadav, Riya Kaushik, Yogendra Singh, Adarsh Kumar” Qualitative Analysis of Text Summarization Techniques and Its Applications in Health Domain”, 09 Feb 2022.

[4] Elena Lloret, “Qualitative Analysis of Text Summarization Techniques and Its Applications in Health Domain”,2011

[5] Subash Voleti, Chaitan Raju, Teja Rani, Mugda Swetha International Research Journal of Engineering and Technology (IRJET), “Text summarization using natural language processing and Google text to speech API”, International Research Journal of Engineering and Technology (IRJET)Volume: 07 Issue: 05 | May 2020

[6]Frederic Kirstein “Automatic text summarization in NLP”, In Proceedings of the 2nd Workshop on Natural Language Generation, Evaluation, and Metrics (GEM), pages 54–77, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

[7] Sebastian Gehrmann, Yuntian Deng, Alexander M. Rush “Bottom-Up Abstractive Summarization”, Volume: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Month: October-November, Year:2018

[8] Ganest and Lapalme, “A Study on Abstractive Summarization Techniques in Indian Languages”, Fourth International Conference on Recent Trends in Computer Science and Engineering, Procedia Computer Science 87 (2016) 25 – 31.

[9] Syed Muqtadir Uddin Hussaini, Faraaz Mohd Khan, Faisal Khan, Dr. Abdul Subhane, “Text Summarization using Natural Language Processing”, Journal of Engineering Science, April 2020.

[10]Samrat Babar, “Text Summarization: An Overview”,21 May 2014.

[11] G. Vijay Kumar, Arvind Yadav, B. Vishnupriya, M. Naga Lahari, J. Smriti, D. Samved Reddy “Text Summarizing Using NLP”,2021 The authors and IOS Press. This article is published online with Open Access by IOS Press and distributed under the terms of the Creative Commons Attribution Non-Commercial License 4.0 (CC BY-NC 4).

[12] N. Moratanch, Dr. S. Chitrakala “A Survey on Abstractive Text Summarization”,2016 International Conference on Circuit, Power and Computing Technologies [ICCPCT].

[13] “ news_summary” dataset:https://www.kaggle.com/datasets/sunnysai12345/news-summary