IJCRT.ORG ISSN : 2320-2882
INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)
An International Open Access, Peer-reviewed, Refereed Journal

Comparative Analysis of Abstractive Text Summarization Techniques

Saniya Nande, Punya Nair, Nandana Anil, Venkatkrishnan Naidu

SIES Graduate School of Technology, Nerul, Navi Mumbai.

Abstract: We are inundated with so much info in today's society that it is difficult to sort through it all. This causes documents to become erroneous and crucial information to be lost. It is practically hard to keep track of everything precisely due to the fast-increasing amount of information available; this is known as the "Information Overload" problem. We must figure out how to efficiently summarize all of this data in order to address this difficulty. However, choosing the optimal algorithm for a given task might be challenging given the wide variety of available options. To make this decision easier, our study compares and examines the most popular text summarizing methods currently on the market. We intend to analyse these methods in order to highlight their advantages and disadvantages so that individuals can more easily determine which approach best meets their requirements. In a world where information is always being presented to us, it is imperative to simplify complex material, and having the appropriate tools to do so can be quite beneficial. Therefore, we hope that our Comparative Analysis of Text Summarization Techniques will offer insightful information about the most effective and efficient ways to summarise huge amounts of text.

Keyword : Text summarization, comparative analysis, Web application.

I. Introduction

The enormous amount of textual material in today's information-driven environment makes understanding and analysis extremely difficult. Abstractive text summarization is a key strategy for efficiently condensing large volumes of data into brief and insightful summaries. Abstractive summarization makes long texts shorter while maintaining important context, which improves accessibility and speeds up decision-making. Furthermore, the idea of comparing summarization algorithms comes from the necessity of closely examining and comprehending the subtle differences between different methods used in this field. By using a comparison approach, we can determine the advantages and disadvantages of several algorithms, which helps us choose the best strategy for a certain task. Thus, the goal of this study is to compare and evaluate several abstractive text summary techniques in a methodical manner. We seek to offer thorough insights into each algorithm's effectiveness by utilizing stringent assessment measures like cosine similarity, ROUGE score, and BLEU score. Ultimately, we hope to further the field of text summarization technology.

II. Literature Review: Limitation Existing system or Research gap

Determining the Restrictions of the Existing Abstractive Text Summarization Systems: Insufficient research into significant issues with frequent pattern mining. In-depth analysis of the CHARM algorithm's drawbacks and potential improvements, neglect to look at potential issues or flaws with LDA-based document models for ad hoc retrieval, improper use of punctuation, hyphens, and stop words in language. Summarising information from many forms and sources while adhering to user requirements can be challenging, difficult reading long texts or texts written in several languages. The dependability and cost of employing human summarizers are drawbacks. Annotated datasets, gold standard datasets, and better tools are required for Indian languages, and need for more efficient methods to manage the massive volume of data produced every day.

Analysis of Research Gaps in Existing Literature: Insufficient investigation and advancement in Arabic text summarization. Insufficient focus on summarising multilingual texts. Insufficient investigation of machine learning techniques for text summarization. Comparing abstractive text summarising (ATS) to extractive text summarising (ETS), there is insufficient research on the former. There hasn't been much progress in creating algorithms that can provide summaries similar to what humans can. Talk on the Need for a Comparative Analysis:In order to address the noted limitations and research gaps, a comparison study is necessary.It can offer perceptions into the functionality, scalability, and user-friendliness of algorithms on various platforms. Researchers can determine the advantages, disadvantages, and potential areas for development of abstractive text summarising systems by contrasting different methods.Such an investigation can direct future lines of inquiry and propel the field of text summarization technology forward.

III. Proposed System

The following explains the main parts and procedures of the system are included in the overarching architecture, or framework. First, the system starts a web data scraping procedure to collect pertinent textual data from online sources after receiving a keyword input from the user. After that, the data is entered into the system, which has an option to add a reference summary to give further background information about the keyword.

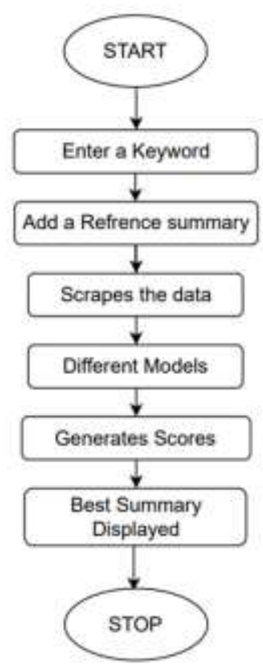

Fig.1 Overall architecture

Methodology: The Fig.2 explains the methodology, here we have five distinct abstractive text summarization algorithms are applied to the scraped data and reference summary when they are obtained: BART, PROPHET NET, FALCON-AI, T5, and PEGASUS. Every algorithm uses a different strategy for summarising, producing summaries according to the incoming data. After summarising, the algorithm uses performance indicators including the cosine similarity matrix, ROUGE score, and BLEU score to assess the output summaries. The efficacy and quality of each algorithm's output are quantified by these metrics. The system uses charts, graphs, and matrices to visually portray the evaluation results in order to make comparison and analysis easier. This makes it simple for consumers to evaluate and contrast how well each algorithm performs. Ultimately, the system determines which algorithm generates the best summary based on the evaluation scores and presents this summary to the user. With the help of this general design, comparative abstractive text summarising is ensured to be methodical and thorough, giving consumers access to high quality summaries that are customised to their input keywords.

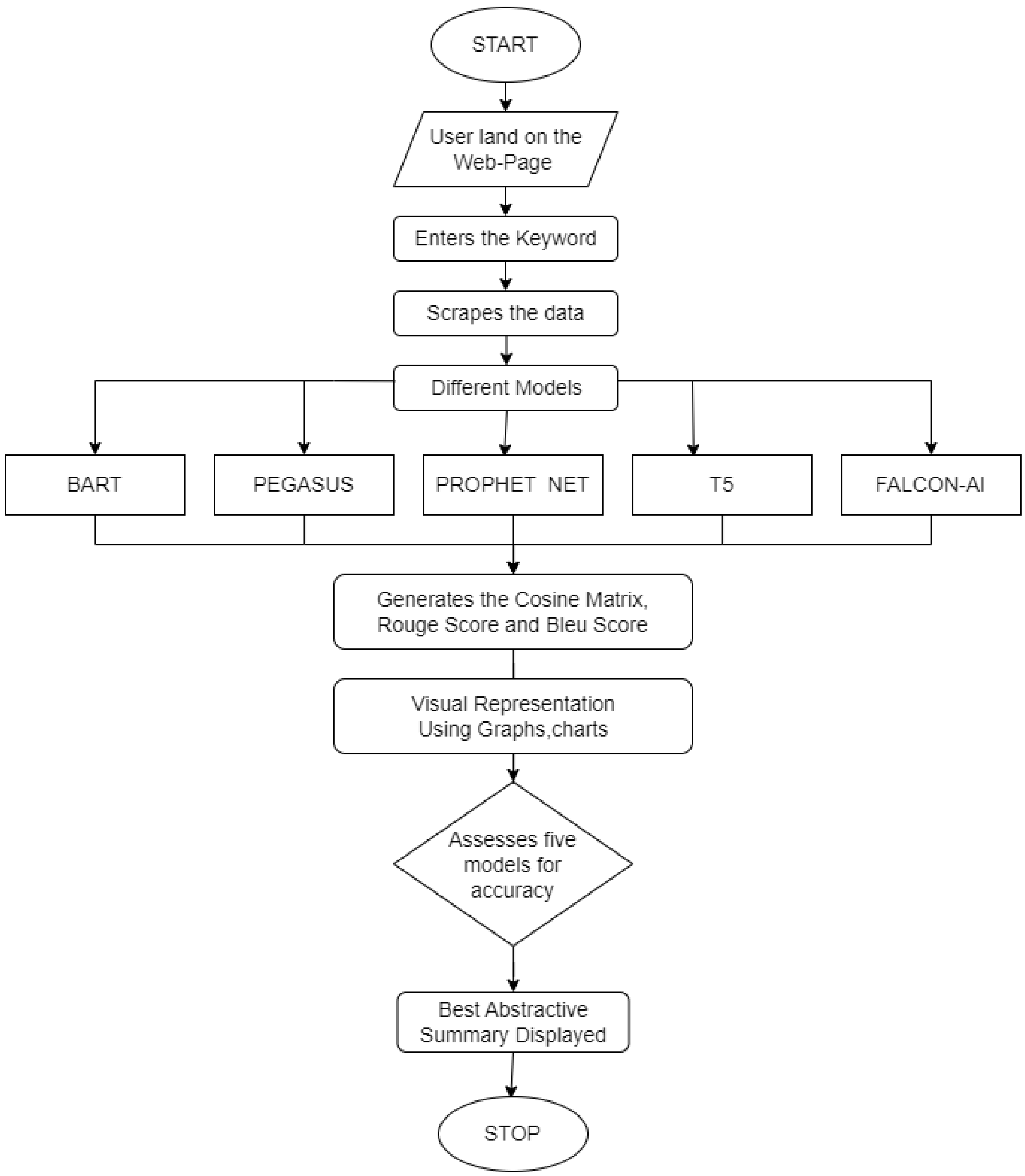

Fig.2 Overall Methodology

Algorithm and Process Design: Five significant abstractive algorithms are known to exist for summarization tasks: BART, PROPHET NET, FALCON-AI, PEGASUS, and T5. Large amounts of information can be condensed into brief summaries using these techniques. Their input is made up of a variety of textual information that has been collected from websites that provide pertinent data. But every algorithm works under a distinct set of rules and reasoning, utilising special techniques to generate summaries that most effectively capture the spirit of the original data. These algorithms employ advanced processing techniques to scrutinise and extract pertinent data, culminating in the creation of summaries that are consistent with their individual methodologies and goals. Let's discuss these algorithms in detail,

1) BART

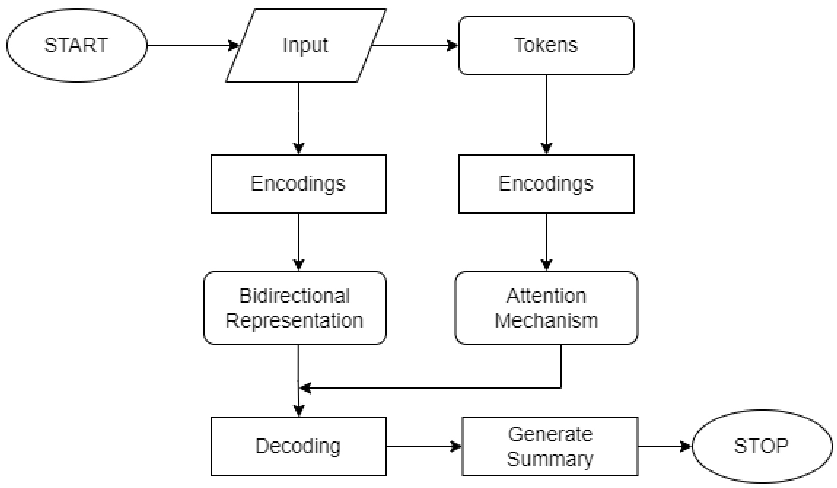

Fig.3 BART Model flowchart

Fig.3 elaborates the working of the BART model. We will look into the steps for the same:

Start: The process of the BART abstractive algorithm is initiated.

Text input: It accepts text input in the form of documents, articles, or any other type of content that needs to be summarised.

Tokens: The input text is tokenized by BART, which divides it into smaller pieces known as tokens. Individual words or subwords from the input text are represented by these tokens.

Encoding: The transformer-based architecture is then used to encode the tokenized input text into a fixed-dimensional representation. The input text's semantic and contextual information is captured during this encoding step.

Bidirectional representation: which allows it to comprehend context in both left-to-right and right-to-left orientations. The model is better able to capture the complex links between words and phrases in the input text thanks to this bidirectional representation.

Attention Mechanism: BART uses attention processes to concentrate on pertinent portions of the input text when creating summaries, both during training and inference. The model's ability to pay attention enables it to give certain terms in the input varied weights, highlighting the data that is most crucial for the process of creating summaries.

Decoding: BART decodes the bidirectional representation created after the input text has been encoded in order to produce the summary. On the basis of the encoded input and previously created tokens, it guesses each token in the summary sequence.

Create Summary: BART creates an abstractive summary of the input text by utilising the attention mechanisms and decoded representation. Concise and comprehensible summary produced by fusing learnt knowledge with information from the incoming text.

Stop: The summary has been produced by the BART abstractive algorithm.

2) PEGASUS

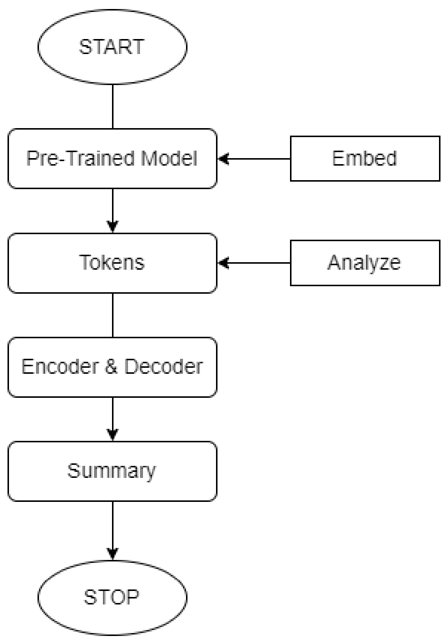

Fig.4 PEGASUS Model flowchart

Fig.4 elaborates the working of the PEGASUS model. We will look into the steps for the same.

Pre-trained Model: PEGASUS establishes a strong foundation for language understanding by using a pre-trained model that has been thoroughly trained on a variety of text data with denoising aims. The model gains a sophisticated understanding of language structures and semantics from this pre-training.

Embed & Tokenize: PEGASUS embeds and tokenizes the text input as it is fed into the system. To enable effective analysis by the model, this procedure entails breaking the text up into smaller pieces (tokenization) and turning words or subwords into dense numerical representations (embedding).

Analysis: After the input has been embedded and tokenized, PEGASUS's encoder carefully examines each token in the text to extract important characteristics and background data. The model can understand the complex linkages and nuances found in the text thanks to this analysis.

Decoder: The abstractive summary is produced by PEGASUS's decoder, which is equipped with the knowledge obtained from the encoder's analysis. The decoder uses a variety of advanced approaches to leverage the encoded representations of the input text and create a summary that captures the key ideas and points of the original text in a clear and concise manner.

Summary Generation: PEGASUS uses attention processes to prioritise important information and concentrate on relevant aspects during the summary generation process. This guarantees that the summary that is generated preserves coherence and conciseness while successfully capturing the essential ideas of the original text.

3) PROPHET NET

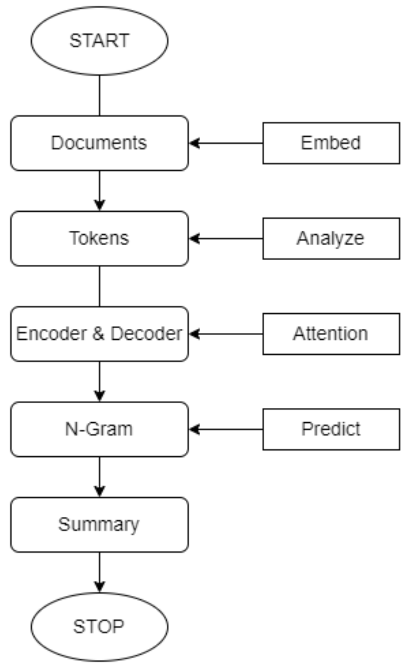

Fig.5 PROPHET NET Model flowchart

Fig.5 elaborates the working of the PROPHET NET model. We will look into the steps for the same

Documents: It uses text passages or documents as its source material for summarising.

Embed & Tokens: The input documents undergo tokenization and embedding, which breaks the text into smaller sections for analysis and turns words into numerical representations.

Analysis: The tokenized input is examined by PROPHET NET's encoder and decoder. As the decoder gets ready to provide the summary, the encoder collects important features and contextual data.

Attention Mechanism: During analysis and summary creation, the model uses attention processes to concentrate on pertinent portions of the input text. This enables it to effectively balance the significance of various words and phrases.

N-gram: To determine the most likely word sequences for the summary, PROPHET NET uses n-gram prediction techniques. This makes the resulting summaries more coherent and fluid.

Predict & Summary Generation: The decoder produces the abstractive summary based on the n-gram predictions, attention processes, and encoded representations. It creates a succinct and logical synopsis by combining the data from the input papers.

4) T5

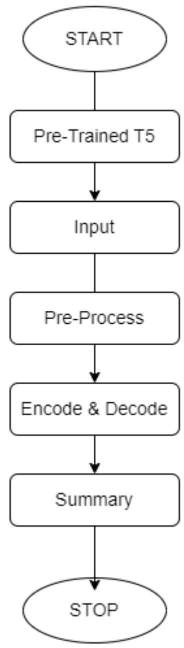

Fig.6 T5 Model flowchart

Fig.6 elaborates the working of the T5 model. We will look into the steps for the same

Pretrained T5: T5 makes use of the Transformer-based T5 (Text-to-Text Transfer Transformer) model, which has been pre-trained. Summarization is one of the many text-to text tasks that this model has been trained on. Text that needs to be summarised, such as paragraphs, articles, or papers, makes up the input for T5.

Preprocess: T5 pre-processes the input to make sure it is compatible with the input format of the model before processing it. Tokenization, segmentation, and other preprocessing procedures might be required for this.

Encode & Decode: The T5 encoder converts the input text into a fixed-dimensional representation. The abstractive summary is simultaneously produced by the T5 decoder using this encoded representation.

5) FALCON-AI

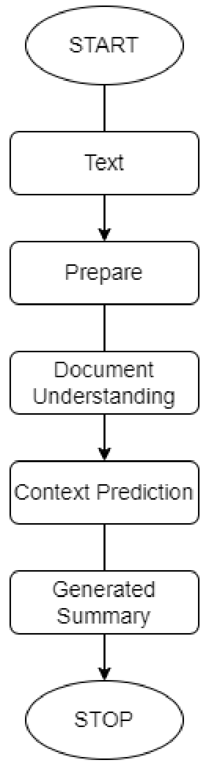

Fig.6 FALCON-AI model flowchart

Fig.6 elaborates the working of the FALCON AI model. We will look into the steps for the same

Start: The abstractive algorithm of FALCON-AI initiates the summarization process.

Text: FALCON-AI receives an input text, which can be an article, document, or any other type of content that has to be summarised. Get ready: FALCON-AI organises, tokenizes, and cleans the input text before processing it to make it ready for analysis.

Document comprehension: To interpret the supplied text, FALCON-AI makes use of sophisticated document comprehension techniques. This entails locating crucial material, recognising essential ideas, and comprehending the document's structure and context.

Context Prediction: FALCON-AI determines the most pertinent information for summarization and forecasts the context based on its comprehension of the incoming text. This phase ensures that the important aspects of the document are captured in the summary that is prepared.

Generated Summary: FALCON-AI creates an abstractive summary of the input text by using the information that has been extracted and the predicted context. It creates a succinct and logical synopsis by combining the essential details and concepts from the text.

Stop: After completing the summarization assignment, FALCON-AI produces a well-written summary that captures the important details from the input text.

IV. Experiment and Results for Validation and Verification.

Results of the project along with their work is displayed below: -

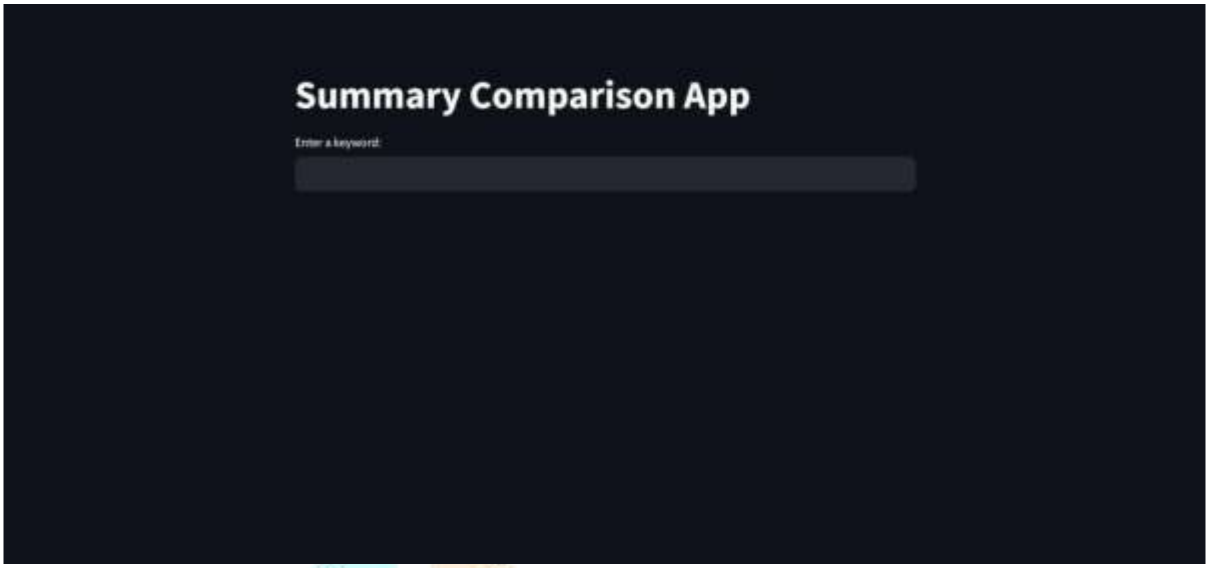

Fig.7 Web-app launched.

Fig.7 explains that the web app is accessible via any browser.

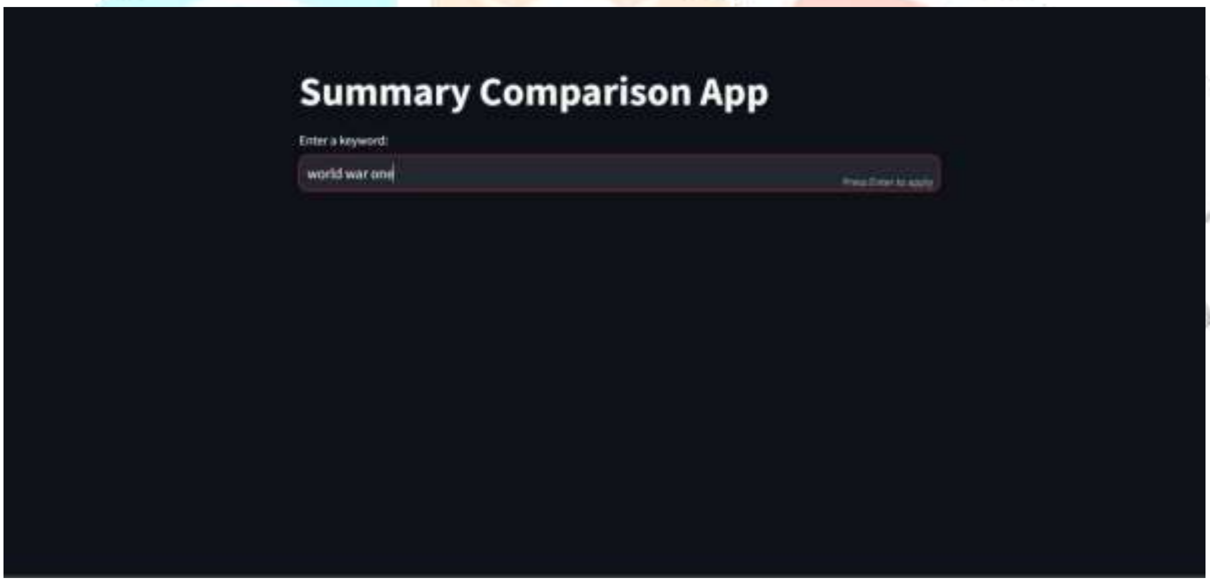

Fig.8 Keyword Entered.

Fig.8 explains that the user can input any keyword into the designated space.

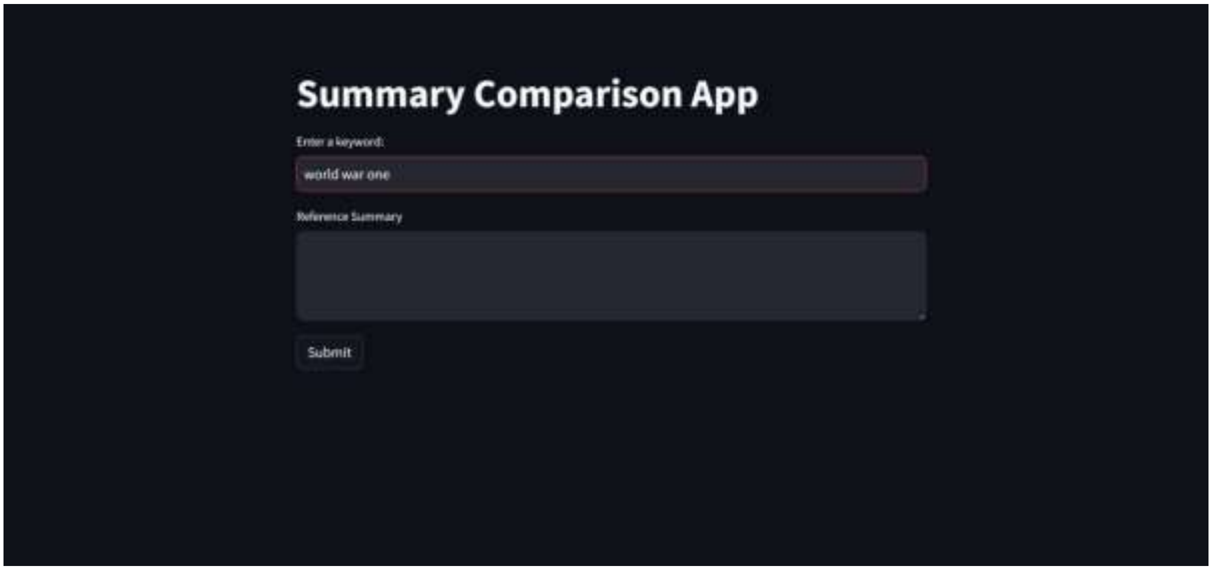

Fig.9 Reference Summary Section.

Fig.9 elaborates that the users must input a reference summary related to their entered keyword, here the entered keyword is World War One.

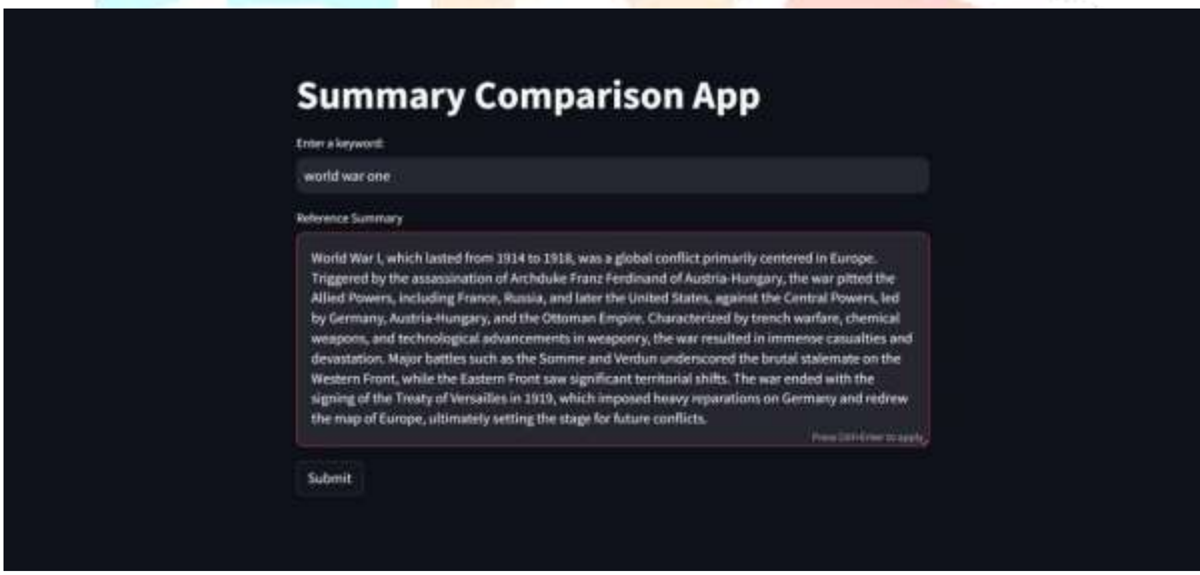

Fig.10 Reference Summary added.

Fig.10elaborates that the user has successfully submitted the reference summary, here it is in regards with World War One.

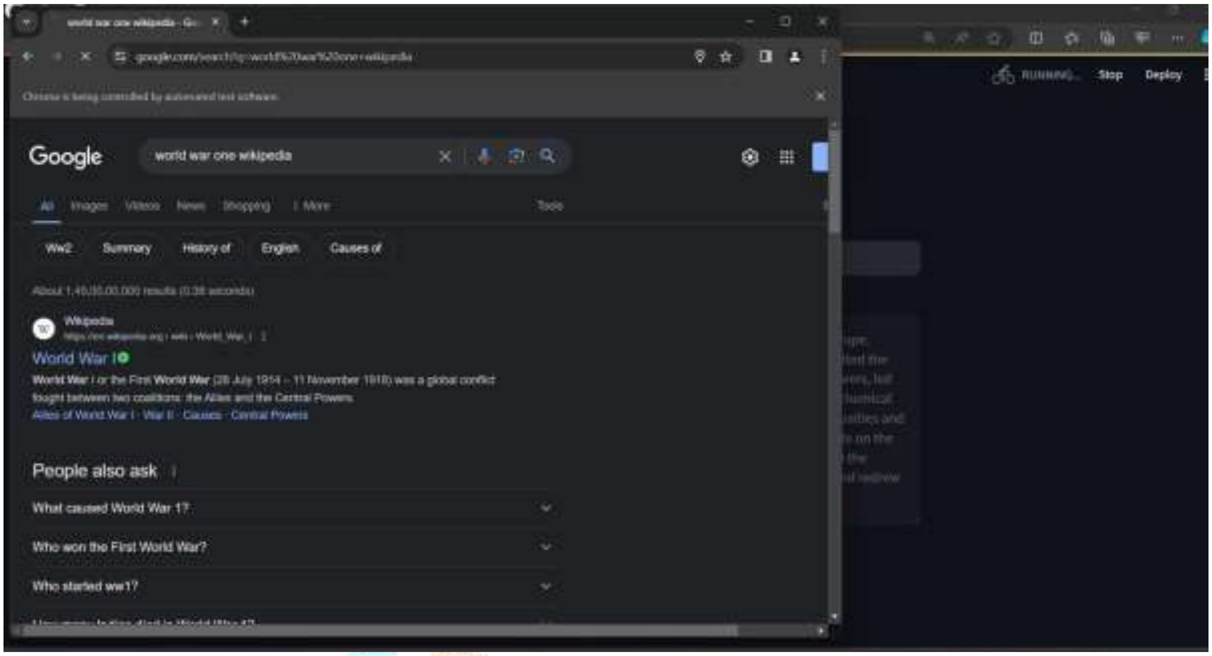

Fig.11 Initial Relevant Data online available.

Fig.11 elaborates that the project scrapes web data based on user-entered keywords.

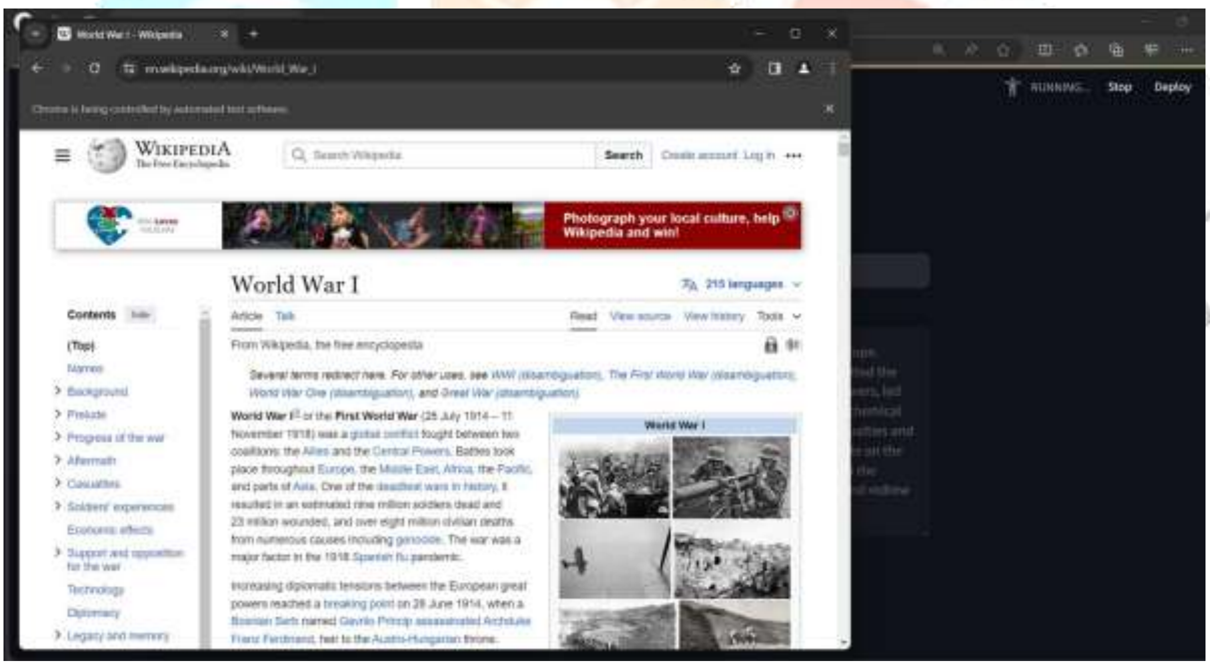

Fig.12 Web scraping from wikipedia.com.

Fig.12 tells us that it is scraping the data from the Wikipedia Website.

V. RESULT

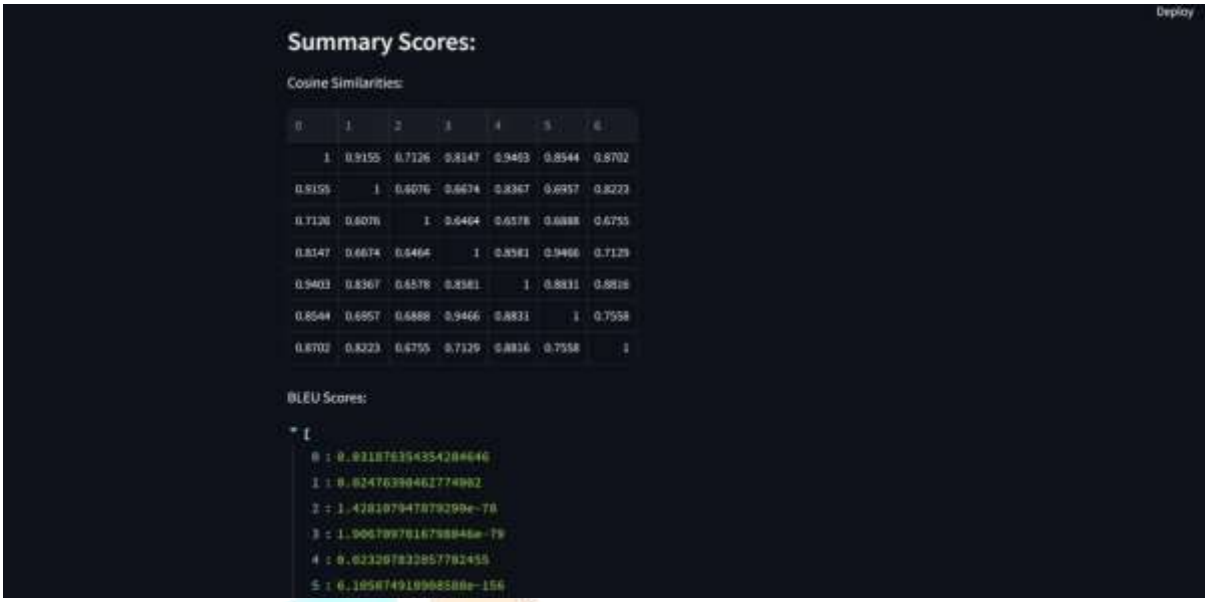

Fig.13 Summary scores generated.

Fig.13 elaborates that the project incorporates multiple summary scores: cosine similarity, BLEU score, and ROUGE score. In summarization, the cosine similarity matrix compares the feature vectors of text summaries to determine how similar they are. It facilitates the evaluation of summarising strategies by determining how comparable or related various summaries are to one another.

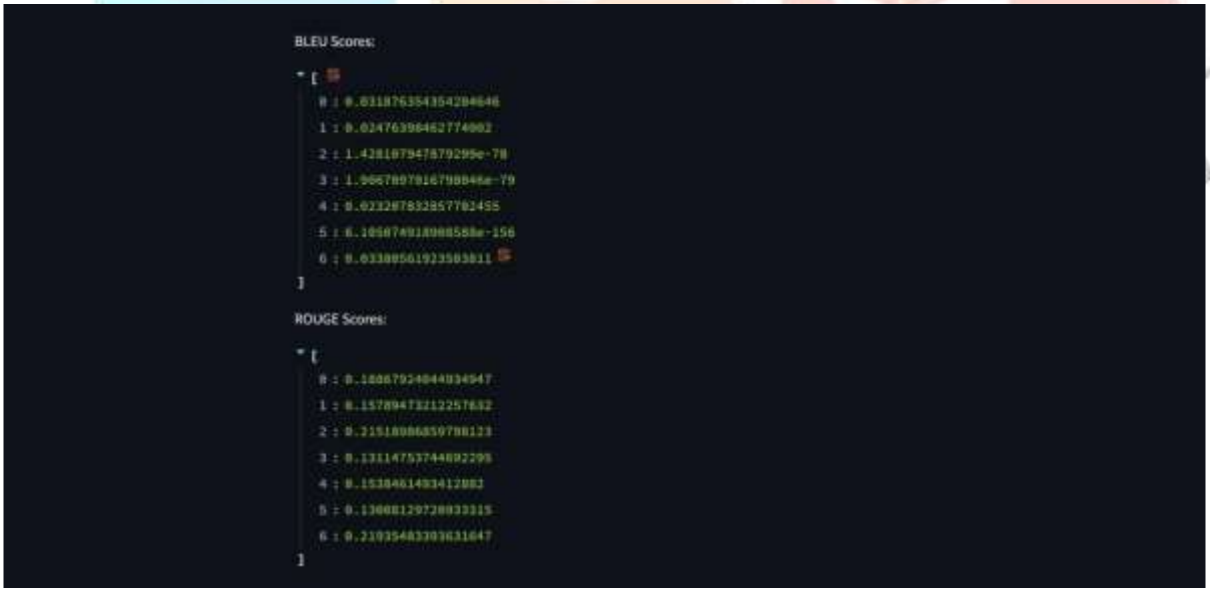

Fig.14 BLEU & ROUGE scores.

Fig.14 elaborates that the BLEU score determines the degree to which the generated summary and the reference summaries match. Better matches are indicated by higher scores. The ROUGE score calculates how closely the generated summary and reference summaries coincide. Better overlap is indicated by higher scores.

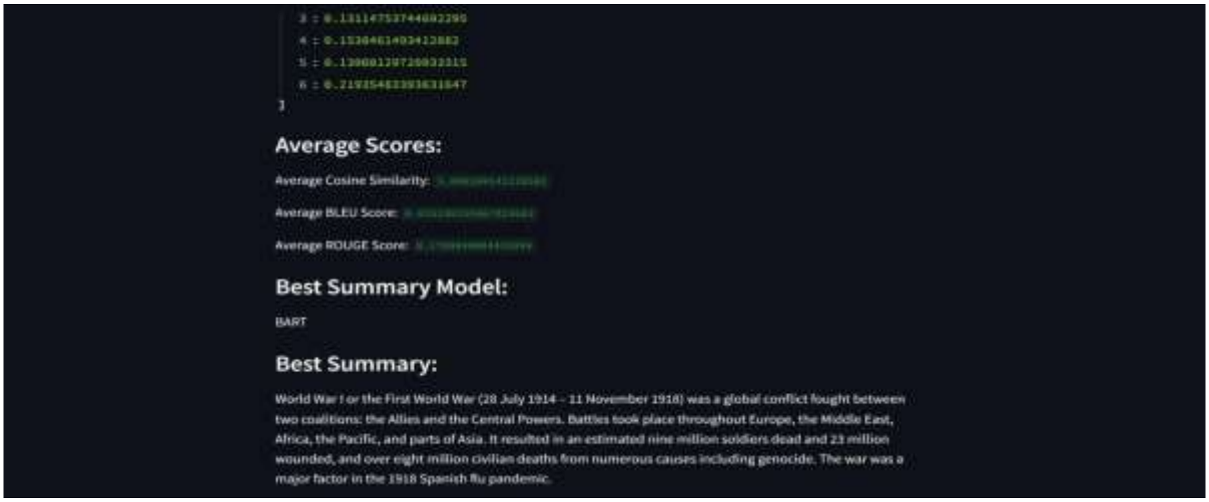

Fig.15 Average scores of COSINES, BLEU & ROUGE.

Fig.15 elaborates that the Text summarization approaches are assessed using conventional metrics, including the average scores of cosine similarity, BLEU score, and ROUGE score. The similarity between produced summaries and reference summaries written by humans is measured. Better alignment and quality of the generated summaries are indicated by higher scores.

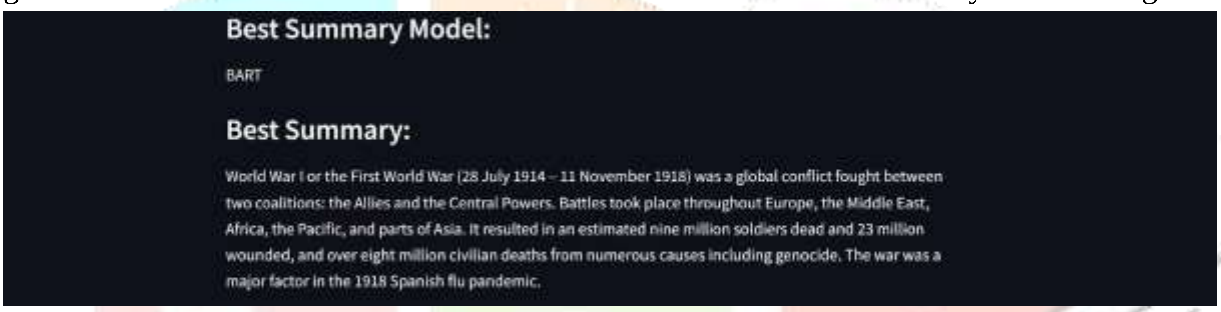

Fig.16 Best Summary model and the Generated Summary.

Fig.16 elaborates that the users are asked to examine different generated summaries and rank the best one next to the most efficient model in the provided text.

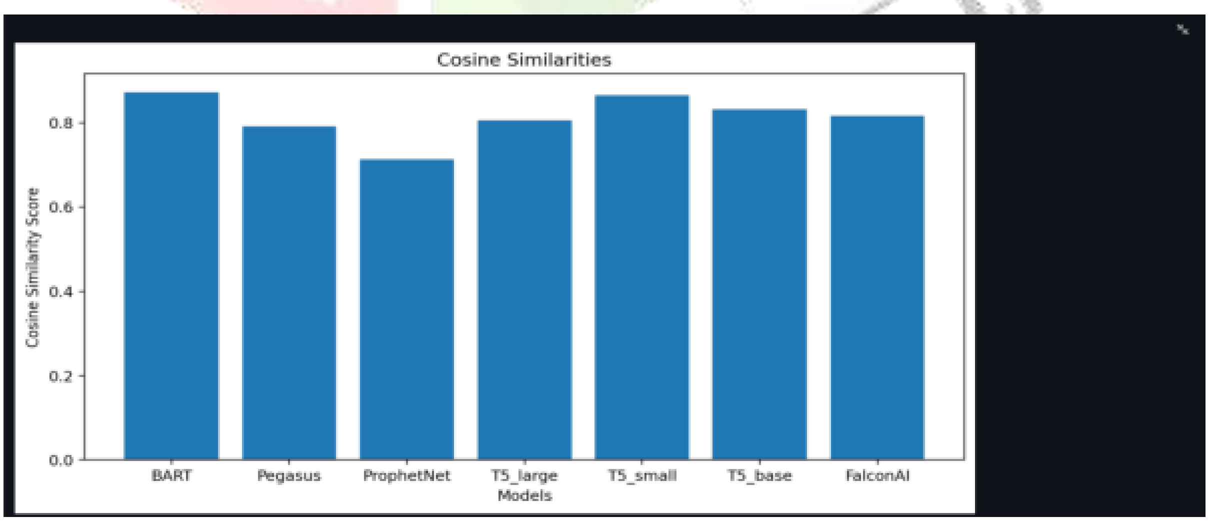

Fig.17 Graphical representation of COSINE SIMILARITY.

Fig.17 explains the Cosine similarity ratings for models such as T5, Prophet Net, Falcon AI, Pegasus, and BART are shown on a graph. By comparing how well their summaries resemble the original text, it facilitates the evaluation of their natural language processing capabilities.

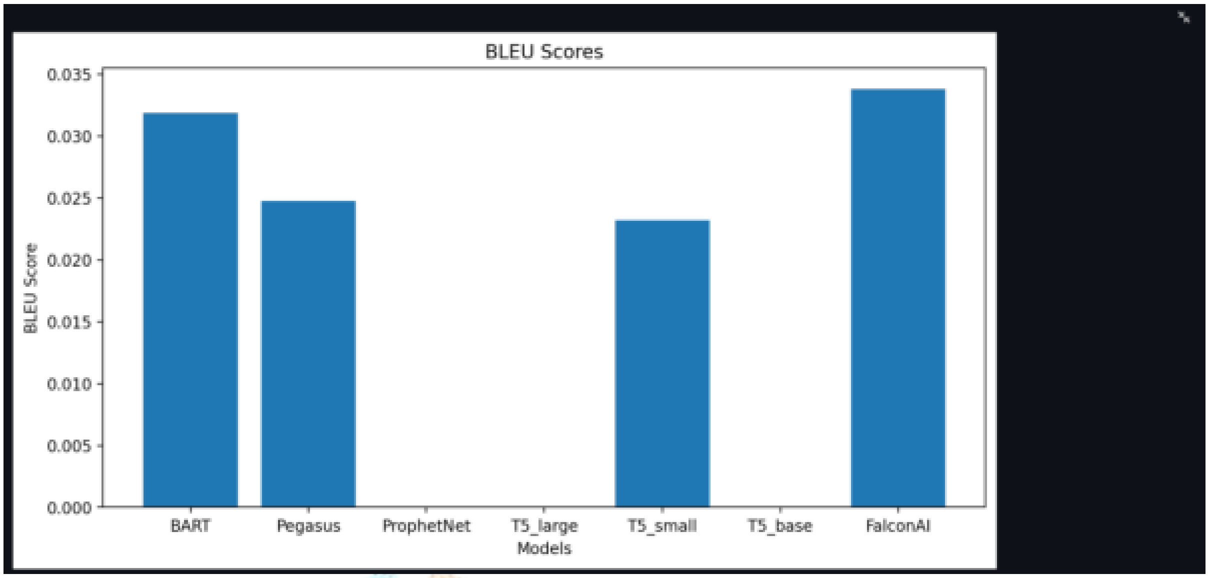

Fig.18 Graphical representation of BLEU SCORE.

Fig.18 explains the BLEU scores for the following models are shown graphically: T5, ProphetNet, Falcon AI, PEGASUS, and BART. The degree of similarity between generated and reference text is measured by BLEU scores. This comparison helps evaluate how well the models produce accurate and insightful summaries.

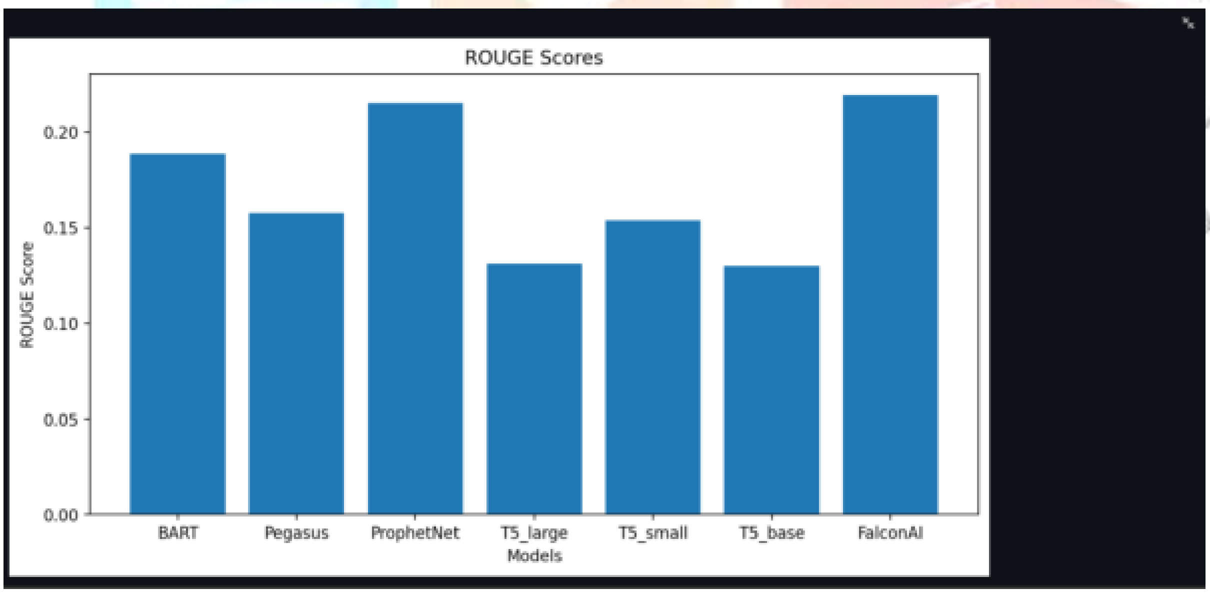

Fig.19 Graphical representation of ROUGE SCORE.

Fig.19 explains the graph that shows the ROUGE scores for the T5, PROPHET NET, FALCON AI, PEGASUS, and BART models, among others. Higher ROUGE ratings denote superior performance, and they quantify the calibre of summaries produced by each model. By analysing this data, it is possible to determine which model generates the most thorough and accurate summaries, which can help with model selection and improvement.

VI. Results & Analysis

The abstractive summarization project's conclusion presents a thorough comparison of five well-known models: T5, PROPHET NET, FALCON AI, PEGASUS, and BART. Of these, the astute assessment has determined that BART is the best model, offering the most sophisticated and thorough summaries. This result highlights the BART model's effectiveness and accuracy in condensing complex data into brief and logical summaries. These results support the importance of methodological rigour and highlight the critical role that sophisticated models like BART play in developing the area of abstractive summarization.

In order to interpret and analyse the experimental results of the comparative abstractive text summarization project, performance measures (cosine similarity, ROUGE score, and BLEU score) for each algorithm must be evaluated, and their advantages and disadvantages must be discussed. Here's one way to go about this analysis:

Comparing Performance Measurements: Based on the generated summaries, compute and compare the cosine similarity, ROUGE score, and BLEU score for each algorithm. Determine which algorithm received the best ratings overall and consider the implications of these results. Take into account any changes or patterns seen in the algorithms' performance across the various measures.

Interpretation of the Results: Determine the overall effectiveness of each algorithm in generating abstractive summaries based on the performance criteria. Examine whether trade- offs exist between measurements or whether certain algorithms regularly outperform others in terms of a range of metrics. When analysing the performance metrics, take into account the generated summaries' quality, coherence, and informativeness.

Discussing the Benefits and Drawbacks: Analyse each algorithm's benefits in relation to how well it produces summaries, including factors such as coherence, fluency, and relevance to the input keyword. Check for any limitations or defects in the algorithms, such as difficulty capturing finer information or generating a range of summaries. Consider how these benefits and drawbacks might impact the algorithms' practical application.

Conclusions and Suggestions: Consider how the experimental findings may affect the field of abstractive text summarization. Provide suggestions for the best algorithm to choose, taking into account the strengths and flaws found, based on certain needs or preferences. Talk about possible directions for further study and development to overcome the shortcomings of current algorithms and enhance overall performance.

VII. Conclusion and Future work

Conclusion: To sum up, the comparative abstractive text summarization effort has shed light on how well five distinct algorithms—BART, PROPHET NET, FALCON-AI, T5, and PEGASUS—perform. By analysing performance indicators such as cosine similarity, ROUGE score, and BLEU score, we have developed a thorough grasp of their advantages and disadvantages when it comes to producing abstractive summaries. Important discoveries include how well some algorithms capture context and produce logical summaries, while others could have trouble maintaining the original meaning or yielding a variety of results. Furthermore, tradeoffs between the generated summaries' relevance, fluency, and informativeness were identified by comparing the metrics.

Future Work: There exist multiple opportunities for future research initiatives and advances in the field of abstractive text summarization going forward.

Investigation of New Algorithms: The quality and efficiency of summaries can be improved by consistently investigating and experimenting with new algorithms or modifications of already-existing ones. It is possible to enhance the effectiveness of summarization models by implementing strategies like reinforcement learning or attention processes.

Enhancement of Evaluation Methodologies: Creating domain-specific evaluation frameworks or adding more extensive metrics to existing evaluation methods might help provide a more nuanced view of the performance of summarization models. This can entail developing new assessment standards that are suited to certain application areas or improving currently available metrics.

Integration of Domain Knowledge: The relevance and accuracy of generated summaries can be increased by incorporating domain-specific knowledge or restrictions into the summarization process. Summarization models can be made more contextual or subject- specific by using strategies like domain adaptation or fine-tuning using domain-specific datasets.

Improvement of User Interaction: Summarization systems can be made more user-friendly and beneficial for end users by improving user interaction features such offering options to customise the summary length, style, or focus on particular content characteristics.

VIII. References

1] S. Bhatia, “A comparative study of opinion summarization techniques,” IEEE Transactions on Computational Social Systems, vol. 8, no. 1, pp. 110–117, Feb. 2021, doi: 10.1109/tcss.2020.3033810.

2] R. S. Sajjan and M. G. Shinde, “A detail survey on automatic text summarization,” International Journal of Computer Sciences and Engineering, vol. 7, no. 6, pp. 991–998, Jun. 2019, doi: 10.26438/ijcse/v7i6.991998.

3] Ç. C. Birant, Ö. Koşaner, and Ö. Aktaş, “A survey to text summarization methods for Turkish,” International Journal of Computer Applications, vol. 144, no. 6, pp. 23–28, Jun. 2016, doi: 10.5120/ijca2016910358.

4] N. Alami, M. Meknassi, and N. Raïs, “Automatic texts summarization: current state of the art,” Journal of Asian Scientific Research, vol. 5, no. 1, pp. 1–15, Jan. 2015, doi: 10.18488/journal.2/2015.5.1/2.1.1.15.

5] Talukder, M.A. et al. (2020) ‘Comparative study on abstractive text summarization’, 2020 11th International Conference on Computing, Communication and Networking Technologies (ICCCNT) [Preprint]. doi:10.1109/icccnt49239.2020.9225657.

6] P. H. Patel, “Pre-Processing Phase of Text Summarization Based on Gujarati Language,” ResearchGate, Jul. 31, 2014. https://www.researchgate.net/publication/264784466

7] S. Yeasmin, P. B. Tumpa, A. Nitu, and M. I. Afjal, “Study of Abstractive Text Summarization Techniques,” ResearchGate, Aug. 24, 2017. https://www.researchgate.net/publication/324438138