GMR Institute of Technology
An Autonomous Institute Affiliated to JNTUK, Kakinada

GMRIT
Training Tomorrow's Engineers Today

A HYBRID APPROACH FOR AUTOMATIC TEXT SUMMARIZATION AND TRANSLATION BASED ON LUHN, PEGASUS, AND TEXTRANK ALGORITHMS

A main project report submitted in partial fulfillment of the requirement for the award of a degree of

BACHELOR OF TECHNOLOGY

In

COMPUTER SCIENCE AND ENGINEERING

Submitted by

Bommuluri Sai Kiran (19341A0519)

Asapu Veera Venkata Vinod Kumar (19341A0513)

Illa Tulasiram (19341A0561)

Basa Venkata Ramana (20345A0504)

Chatradi Raja Venkata Surendra (19341A0533)

Under the esteemed guidance of

Dr. K Srividya

Associate Professor, Dept. of CSE

GMR Institute of Technology

An Autonomous Institute Affiliated to JNTUK, Kakinada

(Accredited by NBA, NAAC with 'A' Grade & ISO 9001:2015 Certified Institution)

GMR Nagar, Rajam – 532127,
Andhra Pradesh, India

November 2022

GMR Institute of Technology
An Autonomous Institute Affiliated to JNTUK, Kakinada

GMRIT
Training Tomorrow's Engineers Today

Department of Computer Science and Engineering

CERTIFICATE

This is to certify that the thesis entitled **A HYBRID APPROACH FOR AUTOMATIC TEXT SUMMARIZATION AND TRANSLATION BASED ON LUHN, PEGASUS, AND TEXTRANK ALGORITHMS** submitted by **B. Sai Kiran (19341A0519), A. V. V Vinod Kumar (19341A0513), I. Tulasi Ram (19341A0561), B. Venkata Ramana (20345A0504), CH.R. Venkata Surendra (19341A0533)** has been carried out in partial fulfillment of the requirement for the award of the degree of **Bachelor of Technology** in **Computer Science and Engineering** of **GMRIT, Rajam** affiliated to **JNTUK, KAKINADA** is a record of bonafide work carried out by them under my guidance & supervision. The results embodied in this report have not been submitted to any other University or Institute for the award of any degree.

Signature of Supervisor
Dr. K. Srividya
Associate Professor
Department of CSE
GMRIT, Rajam.

Signature of HOD
Dr. A. V. Ramana
Professor & Head
Department of CSE
GMRIT, Rajam

The report is submitted for the viva-voce examination held on ………………..

Signature of Internal Examiner

Signature of External Examine

ACKNOWLEDGEMENT

It gives us an immense pleasure to express deep sense of gratitude to my guide **Dr. K. Srividya,** Associate Professor, Department of Computer Science and Engineering for her whole hearted and invaluable guidance throughout the project work. Without her sustained and sincere effort, this project work would not have taken this shape. She encouraged and helped us to overcome various difficulties that we have faced at various stages of our project work.

We would like to sincerely thank our Head of the department **Dr. A. V. Ramana**, for providing all the necessary facilities that led to the successful completion of our project work.

We would like to take this opportunity to thank our beloved Principal **Dr.C.L.V.R.S.V.Prasad**, for providing all the necessary facilities and a great support to us in completing the project work.

We would like to thank all the faculty members and the non-teaching staff of the Department of Computer Science and Engineering for their direct or indirect support for helping us in completion of this project work.

Finally, we would like to thank all of our friends and family members for their continuous help and encouragement.

B. Sai Kiran **19341A0519**
A.V.V. Vinod Kumar **19341A0513**
I. Tulasi Ram **19341A0561**
B. Venkata Ramana **20345A0504**
CH.R.V. Surendra **19341A0533**

ABSTRACT

Nowadays there is a huge demand for text summarization since the primary goal of a text summarising system is to extract the most crucial information from the given text and display it to the end users, there is now a high demand for text summarization tools. In this project, we've developed a web application that can take any broad paragraph as input and, by recognising text features and translating the summarised text into any language, output a condensed form of that specific paragraph. In order to summarise the text, we have presented a hybrid model based on the Pegasus model, an abstractive summary technique, and the Luhn and Textrank algorithms, which are extractive summarization techniques. Based on their ROGUE ratings, this hybrid model was also examined with the BERT, XLNet, and GPT2 models. The translator receives the created ideal paragraph as input and converts the compressed text into any language. In this study, we also aimed to enhance results from the proposed hybrid model in comparison to other models already in use. First, the sentences are ranked according to priority using the text rank algorithm. Secondly, abstractive summarization through using Pegasus model is performed on this paragraph to create a fresh summary with good context, which is then passed on to the Luhn algorithm, which creates the final optimal paragraph. The proposed hybrid model achieved a higher average ROGUE-I score when compared to other existing models such as BERT, XLNet, and GPT2 .

Keywords: *Pegasus, BERT, XLNet, GPT2, abstractive summarization, TextRank, ROGUE, Luhn.*

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS & ABBREVIATIONS (Alphabetic order)

ATS	:	Automatic Text Summarization.
BERT	:	Bidirectional Encoder Representation from Transformers.
BLEU	:	Bilingual Evaluation Understudy.
GPT2	:	Generative Pre-Trained Transformer(second version) .
PEGASUS	:	Pre-training with Extracted Gap sentences for Abstractive Summarization Sequence to Sequence Models.
ROGUE	:	Recall Oriented Understudy for Gisting Evaluation.
TF-IDF	:	Term Frequency-Inverse Document Frequency.
XSUM	:	Extreme Summarization.

1. INTRODUCTION

Text summarization is the act of computationally compressing a piece of data to produce a summary that captures the key ideas or information from the original text. Extractive and abstractive summarising techniques are the two different categories of summarization methods. The traditional methods of extractive summarising have as their primary goal of recognition the key sentences in the text and their inclusion in the summary. This method generates a summary that is identical from the text data's original sentences. The abstractive summarization techniques are the advanced methods, with the approach to identify the important sentences and interpret the context and reproduce the text in a new way. This makes sure that the information's meaning is communicated in the clearest way feasible. Unlike extractive summarization techniques, which simply extract sentences from the raw text data, the summary's sentences are generated by the model.

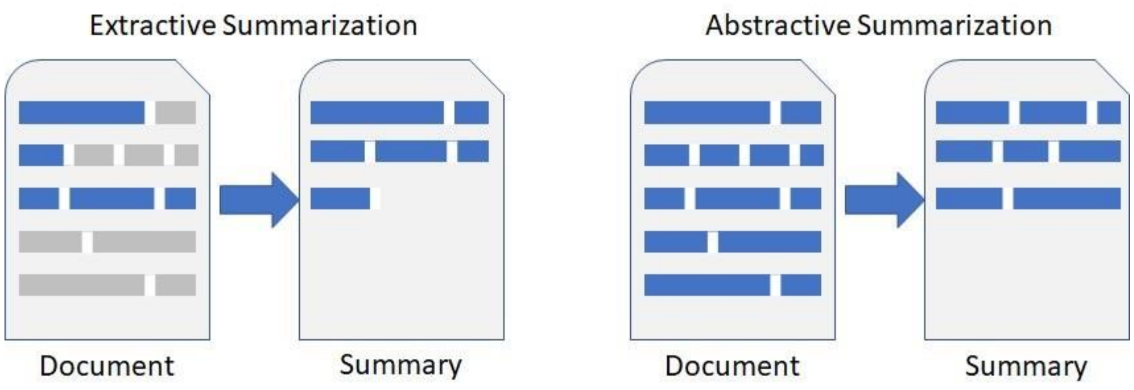

Fig 1.1 Types of Summarization

In this study, we have developed a hybrid model based on the Pegasus model, an abstractive summarization technique, and the Luhn and Textrank algorithms, which are extractive summarization techniques. First, the sentences are ranked according to priority using the text rank algorithm. Then, abstractive summarization using the Pegasus model is done on this paragraph to create a fresh summary with good context, which is then passed on to the Luhn algorithm, which creates the final optimal paragraph. While using the same architecture as BERT, XLNet beats it in many tasks, but when it comes to summarization, XLNet is able to recognise the dependency between any two dependency words or phrases in a much more effective manner, providing much more meaningful sentences than the BERT. GPT2 stands for Generative Pre-Trained Transformer(second version). Its autoregressive characteristics set it distinct from BERT. The primary benefit of GPT2 over BERT is its ability to predict and produce new words to create sentences with a lot more complexity.

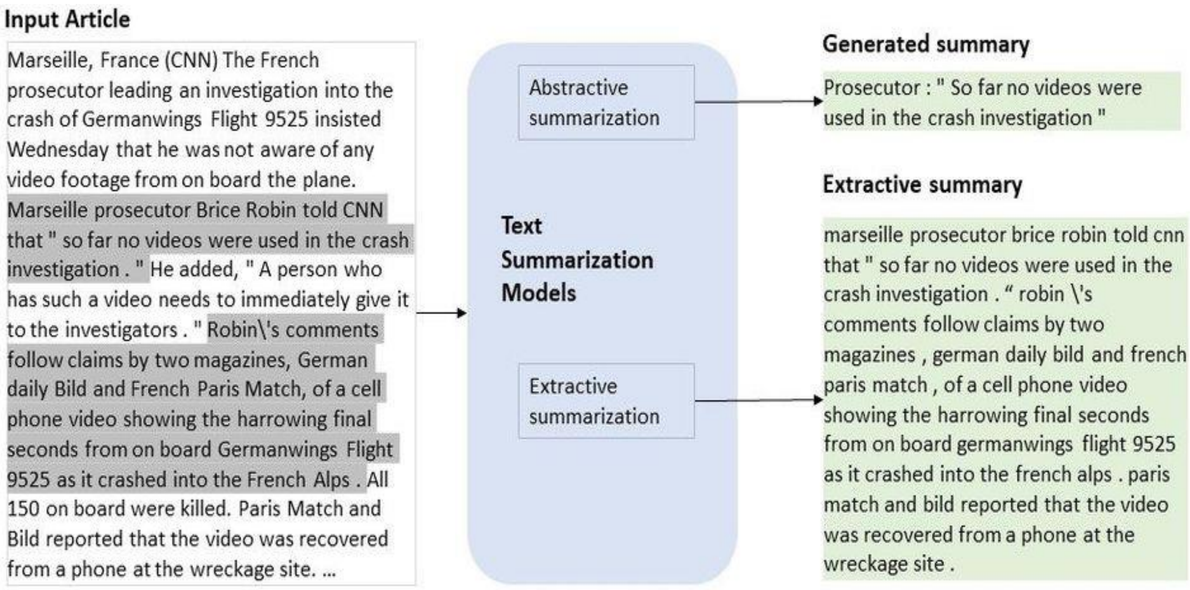

Fig 1.2 Text Summarization

In this project, the proposed hybrid model is compared with the above-mentioned summarization techniques by using the ROGUE scores to evaluate the paragraph generated by all four models. The output summarized paragraph that has been generated by the proposed model is then passed on to a translator and is then translated into any language based on the user input. This is done by using the simple translator package in python.

2. RELATED WORK

This module consists of a literature survey that has been conducted on all the references that have been inferred to propose this model. This module also consists of a comparison table between all the references comparing their methodology, accuracy, and other factors. Based upon this comparison table a hierarchy diagram has been drawn to further simplify the content present in these related works modules understandable.

2.1 Literature Survey

[1]. Ma, T., Pan, Q., Rong, H., Qian, Y., Tian, Y., & Al-Nabhan, N. (2021). T-bertsum: Topic-aware text summarization based on bert. IEEE Transactions on Computational Social Systems, 9(3), 879-890.

In this journal, the author proposed a topic-aware abstractive and extractive text summarization, which is based on BERT. CNN/Daily mail and XSum datasets demonstrate that the proposed model achieves new state-of-the-art results. Stacking the transformer layer in the encoding stage is able to enhance the BERT's ability to represent source texts, make full use of self-attention, and judge the importance of different components of the sentence through different focus scores. The two-stage extractive–abstractive model can share information and generate salient summaries, which reduces a certain degree of redundancy. The ROUGE score of T-Bertsum is 43.85 and the model can generate high-quality summaries with outstanding consistency for the original text but this method has limited processing power for large texts.

[2]. Mrinalini, K., Vijayalakshmi, P., & Nagarajan, T. (2022). SBSim: A Sentence-BERT Similarity-Based Evaluation Metric for Indian Language Neural Machine Translation Systems. IEEE/ACM Transactions on Audio, Speech, and Language Processing, 30, 1396-1406.

In this journal, the author proposed a sentence-BERT-based similarity(SBSim) metric, which is an evaluation metric for machine translation of Indian languages, ie English to Hindi, and English to Tamil Neural Machine Translation systems. In this journal, the proposed SBSim metric, makes use of a BERT model and sentence-level embedding to evaluate Neural Machine Translation outputs. This SBSim metric is compared with the traditional string-based metrics like BLEU, and ChrF++ scores, which are widely used to evaluate MT systems. The proposed metric is also evaluated on the WMT2020 dataset and reports the highest correlation of 0.7129 with the human scores in evaluating outputs from English-to-Tamil and English-to-Hindi NMT systems.

[3]. Wang, Q., Liu, P., Zhu, Z., Yin, H., Zhang, Q., & Zhang, L. (2019). A text abstraction summary model based on BERT word embedding and reinforcement learning. Applied Sciences, 9(21), 4701..

In this journal the author proposed a model for a novel hybrid model of extractive-abstractive to combine BERT (Bidirectional Encoder Representations from Transformers) word embedding with reinforcement learning. The proposed model is compared with the current popular automatic text summary model on the CNN/Daily Mail dataset and uses the ROUGE (Recall-Oriented Understudy for Gisting Evaluation) metrics as the evaluation method. In the future the proposed model can be extended with another pre-training model that is more suitable for the generative task and combines the fine-tuning pre-training model with the abstractive summary task. This model achieved a ROGUE1 score of 37.22 and a ROGUE2 score of 15.78.

[4]. Li, P., Yu, J., Chen, J., & Guo, B. (2021). HG-News: News Headline Generation Based on a Generative Pre-Training Model. IEEE Access, 9, 110039-110046..

In this journal the author proposed a news headline generation model. The generation model is no longer a framework with an encoder-decoder structure. This model works on the NEWS dataset and shows that our model achieves comparable results in the field of news headline generation. In the model with a decoder only, the current token of the target words cannot only focus on the source tokens but also focus on the generated tokens. The decoding process in our model is just like the human reading process which makes our model effective. The proposed model achieved a ROUGE-1 score of 35.8. Further research is to improve the capability of the feature representation and the accuracy of the word generation. The disadvantages of this model include the out-of-vocabulary problem and the word generated by the model sometimes is not correct.

[5]. Gidiotis, A., & Tsoumakas, G. (2020). A divide-and-conquer approach to the summarization of long documents. IEEE/ACM Transactions on Audio, Speech, and Language Processing, 28, 3029-3040.

In this work, a novel divide-and-conquer method for the neural summarization of long documents. The proposed method exploits the discourse structure of the document and uses sentence similarity to split the problem into an ensemble of smaller summarization problems. The proposed model, Dancer breaks a long document and its summary into multiple source-target pairs, which are used for training a model that learns to summarize each part of the document separately. These partial summaries are then combined in order to produce a final complete summary. DANCER is a simple yet effective extension that can boost the performance of different summarization models with minimal additional effort and resources and achieved a good ROGUE-1 score of 45.01.

[6].Akhtar, N., Beg, M. S., & Javed, H. (2019, August). TextRank enhanced topic model for query focussed text summarization. In 2019 Twelfth International Conference on Contemporary Computing (IC3) (pp. 1-6). IEEE.

In this work, a topic model-based summarization method namely the two-tiered topic model is combined with the graph-based TextRank method. The combined method, called TextRank enhanced Two-Tiered topic model, uses the important sentences obtained from TextRank in the generative process of the two-tiered model to extract better summary sentences. The proposed method's summary results outperform other topic model-based summary results using ROUGE metrics evaluated on DUC 2005 dataset. The combined methods TReTTM and TReETTM outperform both TTM and ETTM on Rouge-1 and Rouge-2 evaluation. They also outperform sentence-based models LDCC and SenLDA-based summarization methods.

[7]. Tan, X., Zhuang, M., Lu, X., & Mao, T. (2021). An analysis of the emotional evolution of large-scale internet public opinion events based on the BERT-LDA hybrid model. IEEE Access, 9, 15860-15871

In this journal the author proposed an improved BERT-LDA hybrid model that was constructed in a complex Cantonese context, involving a mixture of Chinese and English, as well as traditional characters and emoticons. Through the collection of large-scale text data related to the Anti-ELAB Movement from a well-known forum in Hong Kong, a BERT-LDA hybrid model for large-scale network public opinion analysis was constructed in a complex context. The analysis and prediction of sentiment evolution of public opinion data, have been attempted to investigate the laws of emotional evolution for such large-scale public opinion events. The improved BERT-LDA model or sentiment classification AUC value exceeds 99.6% in the sentiment classification task for the Anti-ELAB Movement.

[8]. Mridha, M. F., Lima, A. A., Nur, K., Das, S. C., Hasan, M., & Kabir, M. M. (2021). A survey of automatic text summarization: Progress, process and challenges. IEEE Access, 9, 156043-156070.

This journal outlines extractive and abstractive text summarization technologies and provides a deep taxonomy of the Automatic text summarization(ATS) domain. The taxonomy presents the classical Automatic text summarization(ATS) algorithms to modern deep learning Automatic text summarization(ATS) architectures. In this journal, they have also presented a systematic survey of the vast ATS domain in various phases: the fundamental theories with previous research backgrounds, dataset inspections, feature extraction architectures, influential text summarization algorithms, and performance measurement matrices. This journal also presents the current limitations and challenges of ATS methods and algorithms, which can be further used to overcome the limitations in future studies.

[9]. Vathsala, M. K., & Holi, G. (2020). RNN based machine translation and transliteration for Twitter data. International Journal of Speech Technology, 23(3), 499-504.

In this paper the author aims at analyzing the social media data for code-switching and transliterated to English language using the special kind of recurrent neural network (RNN) called Long Short-Term Memory (LSTM) Network. The proposed model is compared with BLEU score obtained for DNN methodology to sequence-to-sequence problems using multi-layered LSTM and proved that methodology not only outperforms SMT-based system but also standard Recurrent Neural Network (RNN) can be easily trained with a greater accuracy. In future The present work can be extended to other social and professional media sites such as Facebook, Instagram, LinkedIn etc and also it can be extended to perform content search associated with improper video, audio and image content posted on social media.

[10]. Bawa, S., & Kumar, M. (2021). A comprehensive survey on machine translation for English, Hindi and Sanskrit languages. Journal of Ambient Intelligence and Humanized Computing, 1-34.

In this paper the author proposed transforming text from one language to another by using computer systems automatically or with little human interventions is known as Machine Translation System (MTS). The purpose of this paper is to present a comprehensive survey of MTS in general and for English, Hindi and Sanskrit languages. Preferred Reporting Items for Systematic Reviews and Meta-Analyses (PRISMA) approach including tools and evaluation methods as done in this survey specifically for English, Hindi and Sanskrit languages. BLEU Scores For English-Sanskrit Machine translation system is 0.445 and English-Hindi Machine Translation System is 0.75**.**

[11]. Ning, J., & Ban, H. (2021). Design and Testing of Automatic Machine Translation System Based on Chinese-English Phrase Translation. Mobile Information Systems, 2021.

In this journal, the author introduces a phrase-based automatic machine translation system by combining machine translation methods with Chinese-English phrase translation and explores the design and testing of machine automatic translation systems. Automatic machine translation is a complete process that integrates the development of concepts, opens up the use of existing resources, and adds modules such as repositories, dictionaries, and so on. The main disadvantage of this model is that it is not reliable as it does not have enough dependency pairs for proper translation. The proposed model achieved a BLEU Score of 13.5 and This model proposed results in a short time with comparable BLEU scores.

[12]. Ke, X. (2022). English synchronous real-time translation method based on reinforcement learning. Wireless Networks, 1-13.

In this paper the author proposed an implementation on the real-time synchronous translation method, and focus on the key technologies to be solved in the translation generation of real-time synchronous translation method. The dataset used in this paper was ChinaDaily dataset and take Recall-Oriented Understudy for Gisting Evaluation (ROUGE) as the evaluation index. In Future, in terms of time the Tri-Trophic Metapopulation Mode(TTMM) system has increased the translation results compared to the Baseline system, which needs to be improved further. The experimental results show that the mixed real-time synchronous translation method and RL brings a certain degree of optimization and achieved a ROGUE1 score of 36.71.

[13].Kano, T., Sakti, S., & Nakamura, S. (2020). End-to-end speech translation with transcoding by multi-task learning for distant language pairs. IEEE/ACM Transactions on Audio, Speech, and Language Processing, 28, 1342-1355.

In this journal, the author proposed a Traditional speech-to-speech translation approach to concatenate automatic speech recognition (ASR), text-to-text machine translation (MT), and text-to-speech synthesizer (TTS) by text information. The results for the RNN-based model in natural speech are slightly worse compared to the performance with generated speech. But, if they have used the Transformer instead of RNN that is trained using both natural and generated speech, a high ASR performance was achieved. This model proposed results in a short time with high BLEU score when compared to other models and achieved a BLEU Score of 34.3.

[14]. Heo, Y., Kang, S., & Yoo, D. (2019). Multimodal neural machine translation with weakly labeled images. IEEE Access, 7, 54042-54053.

In this paper the author proposed a multimodal neural machine translation system that uses both texts and their related images to translate Korean image captions into English. This paper uses data that extends the Flickr30K Entities dataset, where the entities in the image are labeled, each image has its own caption and each of it has a source sentence. The results can be analyzed considering three aspects the performance change corresponding to the Korean input unit, the effect of image features, and the effect of label candidates. The proposed model improved the performance by +1.0 BLEU compared to the text-based NMT model and achieved a BLEU score of 30.7.

[15]. Sen, O., Fuad, M., Islam, M. N., Rabbi, J., Masud, M., Hasan, M. K., ... & Iftee, M. A. R. (2022). Bangla Natural Language Processing: A Comprehensive Analysis of Classical, Machine Learning, and Deep Learning Based Methods. IEEE Access.

In this paper, the author presented an analysis of 75 BNLP research papers and categorize them into 11 categories, namely Information Extraction, Machine Translation, Named Entity Recognition, Parsing, Parts of Speech Tagging, Question Answering System, Sentiment Analysis, Spam and Fake Detection, Text Summarization, Word Sense Disambiguation, and Speech Processing and Recognition The author studied articles published between 1999 to 2021, and 50% of the papers were published after 2015. This journal presents a complete analysis of all the natural language processing methods which can be used to overcome future limitations. At last the author discussed challenges and future research possibilities and further reviewed the characteristics and complexity essential to understanding modern challenges in this field.

[16]. Mallick, C., Das, A. K., Dutta, M., Das, A. K., & Sarkar, A. (2019). Graph-based text summarization using modified TextRank. In Soft computing in data analytics (pp. 137-146). Springer, Singapore.

In this paper, a graph-based text summarization method has been described which captures the aboutness of a text document. The method has been developed using modified TextRank computed based on the concept of PageRank defined for each page in the Web pages. The proposed method constructs a graph with sentences as the nodes and similarity between two sentences as the weight of the edge between them. Modified inverse sentence frequency-cosine similarity is used to give different weightage to different words in the sentence, whereas traditional cosine similarity treats the words equally. The proposed method achieved a ROGUE-1 score of 46.87. The main limitation of the proposed algorithm is that it does not take care of the anaphora resolution problem.

[17]. Zeng, H., & Chen, G. (2020, December). Unsupervised extractive summarization based on context information. In 2020 IEEE 6th International Conference on Computer and Communications (ICCC) (pp. 1651-1655). IEEE.

This paper proposes a model “lead3” of unsupervised extractive summarization. They have tested many ways to express the context information, studied the relationship between sentences in the abstract. It is also proved that the context information and the relationship between the sentences are very helpful to the task and then developed an unsupervised summarization system without any training. The dataset used in this approach is CNN/DM dataset which contains 312,000-word dependency pairs. In this paper the proposed unsupervised extractive summarization model lead3 achieved a ROGUE1 score 40 and ROGUE2 score of 17.

[18].Xie, Q., Bishop, J. A., Tiwari, P., & Ananiadou, S. (2022). Pre-trained language models with domain knowledge for biomedical extractive summarization. Knowledge-Based Systems, 109460.

In this journal, they have proposed KeBioSum, a novel knowledge infusion training framework, and experiment using a number of Pre-Trained Language Models(PLMs) as bases, for the task of extractive summarization of biomedical literature. A novel knowledge-guided training framework, namely the knowledge adapter, was used for both generative and discriminative training to support knowledge infusion into the PLMs. To evaluate the effectiveness of our model, they have conducted experiments on three literature datasets from biomedicine: CORD19, PubMed, and S2ORC. CORD-19 is an open dataset, which includes scientific papers on COVID-19. This PubMed BERT model achieved a decent ROGUE1 score of 42.9 and a ROGUE2 score of 37.0.

[19].Qaroush, A., Farha, I. A., Ghanem, W., Washaha, M., & Maali, E. (2021). An efficient single document Arabic text summarization using a combination of statistical and semantic features. Journal of King Saud University-Computer and Information Sciences, 33(6), 677-692.

In this paper, they have proposed an automatic, generic, and extractive Arabic single document summarizing method aiming at producing a sufficiently informative summary. The proposed extractive method evaluates each sentence based on a combination of statistical and semantic features in which a novel formulation is used taking into account sentence importance, coverage and diversity. Further, two summarizing techniques including score-based and supervised machine learning were employed to produce the summary and then assist in leveraging the designed features. In this paper EASC dataset was taken which comprises of 153 articles. The proposed score-based methods achieved recall, precision, and F-Score of 67.0,61.0,64.0 respectively.

[20]. Iwasaki, Y., Yamashita, A., Konno, Y., & Matsubayashi, K. (2019, November). Japanese abstractive text summarization using BERT. In 2019 International Conference on Technologies and Applications of Artificial Intelligence (TAAI) (pp. 1-5). IEEE

In this paper the author proposed an automatic abstractive text summarization algorithm in Japanese using a neural network. In the proposed a transformer-based decoder returned the summary sentence from the output as generated by the encoder. The dataset used in this paper was a Livedoor news corpus consisting of 130,000 data points, of which 100,000 were used for training and the accuracy of the model was 67%. The contents of the summary sentence were repeated, the model was unable to handle unknown words, and there was a problem with simple word mistakes.

[21].Abdel-Salam, S., & Rafea, A. (2022). Performance Study on Extractive Text Summarization Using BERT Models. Information, 13(2), 67.

This journal proposed a model for text summarization which is BERT and this text summarization is composed of three phases i.e data pre-processing phase, algorithmic processing phase, and post-processing phase. Data pre-processing phase is a process of cleaning the document and the algorithmic Processing Phase is the process of applying an algorithmic approach and the Post-Processing Phase is the process of applying any data transformation to the target summary. The dataset used in this approach is CNN/DM dataset which contains 312,000-word dependency pairs. The training time taken for a DistilBERT summarizer was around 25 minutes per 1000 checkpoints on a Google GPU session and it maintains an accuracy of 98% of the BERT model.

[22].Andrabi, S. A. B., & Wahid, A. (2022). Machine translation system using deep learning for English to Urdu. Computational Intelligence and Neuroscience, 2022.

In this paper Neural machine translation is a novel paradigm in machine translation research. In this paper, an LSTM-based deep learning encoder-decoder model for English to Urdu translation is proposed. The parallel English-Urdu corpus of 1083734 tokens has been used, and out of these total tokens, 542810 were English tokens, and 123636 were Urdu tokens. This model has an average BLEU score of 45.83. In the future, in this model, a speech recognition module can be built with speech-to-text translation. The translation quality of the proposed model was good the word error rates were also less. The main limitation of this model is that it could not translate all the words into the specified language and missed some words.

[23].Gupta, H., & Patel, M. (2021, March). Method Of Text Summarization Using LSA And Sentence-Based Topic Modelling With Bert. In 2021 International Conference on Artificial Intelligence and Smart Systems (ICAIS) (pp. 511-517). IEEE.

In this journal, the author proposed a method of text summarization using LSA and Sentence based topic modeling with BERT. The results in extracting useful sentences from a text document that contains a useful amount of information about the topic on which the text document is based on. The proposed model achieves a ROUGE-1 0.44 score. In the future using the proposed algorithm in abstractive text summarizer where the machine is generating a summary in its own language will result in achieving greater accuracy. The proposed model generates summaries based upon the semantics giving optimal results and performing accurately on large texts.

[24].Srikanth, A., Umasankar, A. S., Thanu, S., & Nirmala, S. J. (2020, October). Extractive text summarization using dynamic clustering and co-reference on BERT. In 2020 5th International Conference on Computing, Communication and Security (ICCCS) (pp. 1-5). IEEE.

In this paper, an existing BERT model is used to produce extractive summarization by clustering the embeddings of sentences by K-Means clustering but in a dynamic method to decide the number of clusters. The dataset used for the summarization task is CNN/DailyMail. The dataset includes CNN and Daily Mail news articles. The pre-trained BERT model has been used in this journal. In future models, variations of BERT should be compared and tested. The ROUGE-1(F1) score is 41.25. The main disadvantage of the existing model was that the entire context of the document to be summarized could not be represented in a smaller number of sentences.

[25].Chen, K., Zhao, T., Yang, M., Liu, L., Tamura, A., Wang, R., ... & Sumita, E. (2019). A neural approach to source dependence-based context model for statistical machine translation. IEEE/ACM Transactions on Audio, Speech, and Language Processing, 26(2), 266-280.

The author proposed a novel neural approach to source dependence-based context representation for machine translation. The proposed model is capable of not only encoding source long-distance dependencies but also capturing functional similarities to better predict translations. The proposed model achieves significant improvement over the baseline systems and outperforms several existing context-enhanced methods. The main limitation of this model is that it has to improve its performance regarding word embeddings and proper translation. The proposed model achieved a descent BLEU Score of 17.8.

[26].Madhuri, J. N., & Kumar, R. G. (2019, March). Extractive text summarization using sentence ranking. In 2019 International Conference on Data Science and Communication (IconDSC) (pp. 1-3). IEEE.

In this paper, a novel statistical method to perform an extractive text summarization on a single document is demonstrated. The method gives the idea of the input text in a short form that is in the form of a meaningful summary. Sentences are ranked by assigning weights and they are ranked based on their weights. Highly ranked sentences are extracted from the input document so it extracts important sentences that direct to a high-quality summary of the input document. The dataset used in this paper is the Stanford sentiment treebank which consists of 10000 reviews from rotten tomatoes segregated based on their polarities. The model achieved a decent F1 score of 62.29.

[27].Chandra, R., & Kulkarni, V. (2022). Semantic and sentiment analysis of selected bhagavad gita translations using BERT-based language framework. IEEE Access, 10, 21291-21315.

In this paper, they have presented a framework that compares selected translations (from Sanskrit to English) of the Bhagavad Gita using semantic and sentiment analyses. They have used a hand-labeled sentiment dataset for tuning state-of-art deep learning-based language models known as bidirectional encoder representations from transformers (BERT). The dataset that has been used in this paper is the SenWave dataset which consists of 10,000 tweets that are hand-labeled by experts and the polarity score varies from 0-10 with respect to positive. The evaluation metric used in this paper is the cosine similarity which achieved a decent score of 62.0.

[28].Xie, Q., Bishop, J. A., Tiwari, P., & Ananiadou, S. (2022). Pre-trained language models with domain knowledge for biomedical extractive summarization. Knowledge-Based Systems, 109460.

In this journal, they have proposed KeBioSum, a novel knowledge infusion training framework, and experiment using a number of Pre-Trained Language Models(PLMs) as bases, for the task of extractive summarization of biomedical literature. A novel knowledge-guided training framework, namely the knowledge adapter, was used for both generative and discriminative training to support knowledge infusion into the PLMs. To evaluate the effectiveness of our model, they have conducted experiments on three literature datasets from biomedicine: CORD19, PubMed, and S2ORC. CORD-19 is an open dataset, which includes scientific papers on COVID-19. This PubMed BERT model achieved a decent ROGUE1 score of 42.9 and a ROGUE2 score of 37.0.

[29].Ramina, M., Darnay, N., Ludbe, C., & Dhruv, A. (2020, May). Topic level summary generation using BERT induced Abstractive Summarization Model. In 2020 4th International Conference on Intelligent Computing and Control Systems (ICICCS) (pp. 747-752). IEEE.

In this paper, the implemented system channels an idea called Topic level summary. The topic level summary is a collective summary in text format which consists of relevant information on a topic where a topic can be an idea, concept or a term user wants to know about. This information in text format is passed on to the abstractive summarization model which uses advanced NLP capabilities of the bidirectional encoder representation transformers (BERT) language model to generate a topic-level summary. The dataset used in this approach is CNN/DM dataset which contains 312,000-word dependency pairs. The mentioned summarization model has ROUGE scores of 41.72, 19.39, and 38.76 for ROUGE-1, ROUGE-2, and ROUGE-L respectively.

[30].Sehgal, S., Kumar, B., Rampal, L., & Chaliya, A. (2019). A modification to graph based approach for extraction based automatic text summarization. In Progress in advanced computing and intelligent engineering (pp. 373-378). Springer, Singapore.

In this paper the author lays emphasis on the TextRank algorithm, a graph-based approach used to tackle the automatic article summarization problem, and proposes a variation to the similarity function used to compute scores during sentence extraction. The TextRank algorithm is based purely on the frequency of occurrence of words and does not require any prior knowledge of grammar. This eliminates the requirement of any particular tools dedicated to any particular language. The proposed model achieved an average recall of 1.0 and a precision of 0.49 and a fscore of 0.61.

Table 2.1 Comparison Table

Sl. No	**TECHNIQUES (i.e. authornComparisoneference numbers)**	**YEAR**	**DESCRIPTION**	**LIMITATIONS**	**ADVANTAGES**	**PERFORMANCE METRICS**	**GAPS**
1.	Tinghuai Ma, Qian Pan, Huan Rong, Yurong Qian, Yuan Tian, and Najla Al-Nabhan.	2022	In this journal, the author proposed a topic-aware extractive and abstractive summarization model named T-BERTSum, based on BERT.	For long articles with multiple topics, the proposed model has limited processing power.	The model can generate high-quality summaries with outstanding consistency for the original text.	ROGUE1- 43.06 ROGUE2- 19.76 ROGUE L- 39.43	Future work will be conducted to capture multiple topics which are much closer to the original text and further prove the validity of the proposed model.
2.	K. Mrinalini, P. Vijayalakshmi, and T.Nagarajan.	2022	In this journal, the author proposed an SBSim metric, that makes use of a BERT model and sentence-level embedding to evaluate NMT outputs.	The main limitation of this model is that it can only work on two language pairs.	The proposed SBSim metric achieves the highest correlation in evaluating outputs from English-to-Tamil and Hindi NMT systems.	SBSim- 0.99.	Further research will be conducted to make this model run for multiple language pairs.

3.	Qicai Wang, Peiyu Liu, Zhenfang Zhu, Hongxia Yin 1, Qiuyue Zhang and Lindong Zhang.	2019	In this journal, the author proposed a novel hybrid model of extractive-abstractive to combine BERT word embeddings with reinforcement learning.	The main disadvantage of this model is that it cannot produce the best summary as the context of the summary was less.	The model proposed in this paper achieves the best results in the CNN/Daily Mail dataset.	ROGUE1- 37.22 ROGUE2- 15.78 ROGUE L- 33.90	Future works include another pre-training model that is more suitable and combines the model with the abstractive summary task.
4.	Ping Li, Jiong Yu, Juaying Chen, and Binglie Guo.	2021	In this journal, the author focuses on news headline generation based on a generative pre-training model.	The disadvantages of this model include the out-of-vocabulary problem and the word generated by the model sometimes is not correct	This model achieves comparable results when compared to other models in news generation.	ROGUE1- 37.19 ROGUE2- 17.46 ROGUE L- 33.71	Further research is to improve the capability of the feature representation and the accuracy of the word generation.
5.	Alexios Gidiotis and Grigorios Tsoumakas.	2020	In this work, a novel divide-and-conquer method also called as dancer was proposed for the neural summarization of long documents.	The main disadvantage of this model is that it can only work with a few pre-trained models.	It is a simple yet effective extension that can boost the performance of different summarization models with minimal additional effort and resources.	ROGUE-1 – 45.01.	Future work will be to combine DANCER with more complex summarization models that could improve summarization quality.
6.	Nadeem Akhtar, M M Sufyan Beg, Hira Javed.	2019	In this work, a topic model-based summarization method namely the two-tiered topic model is combined with the graph-based TextRank method.	The main limitation of this model is that it could not be integrated with complex neural networks.	This model can make use of both topic model based and graph based approaches for query focused summarization.	Recall – 0.36. F Measure – 0.34. Precision – 0.35.	Further research will be to use different form of sentence graphs which csn be used to find sentence TextRank scores.
7.	Xu Tan, Muni Zhuang, Xin Lu, and Taitian Mao.	2021	In this journal, the author proposed an improved BERT-LDA hybrid model that was constructed in a complex Cantonese context for sentiment analysis.	The main disadvantage of this model is that it focused on only a single topic and could only be performed on shorter texts.	This model proposed results in a short time with high accuracy and low error rates of less than 9.95%.	NPMI Value- 0.703	

8.	M.F. Mridha , Aklima Akter , Kamruddin Nur, Sujoy Chandra Das, Mahmud Hasan, and Muhammad Mohsin Kabir.	2021	This journal, outlines extractive and abstractive methods and provides an idea on Automatic Text Summarization.		This journal provides a brief survey on Automatic text summarization methods and algorithms which can be used to overcome problems.	Metrics used are ROGUE scores, F-Score, and Accuracy of different models.	Further research will be conducted to overcome the limitations in this journal.
9.	M. K. Vathsala, Holi Ganga.	2020	The author proposed a model by combining RNN and LSTM for Twitter data translation.	The main disadvantage of this model is that it could find proper dependency pairs for few words thus affecting its accuracy.	This model can be used to perform transliteration and also to improve security and restrict sensitive content on social media.	BLEU Score – 0.13	Future work will be conducted to perform a content search in the form of audio, video, images, etc.
10.	Sitender, Seema Bawa, Munish Kumar, Sangeeta.	2021	The purpose of this journal is to present a comprehensive survey of MTS in general and for English, Hindi, and Sanskrit languages in particular.	The English to Sanskrit machine translation system did not perform well because Sanskrit is a complex language.	This journal proposes a survey of machine translation systems that helps to overcome any problems related to MTS.	BLEU Scores For English-Sanskrit MTS- 0.445. English-Hindi MTS- 0.75.	Further research will be conducted to further improve the MT systems and in the development of new MTS.
11.	Jing Ning and Haidong Ban.	2021	In this journal, the author introduces a phrase-based automatic machine translation system by combining machine translation methods with Chinese-English phrase translation.	The main disadvantage of this model is that it is not reliable as it does not have enough dependency pairs for proper translation.	This model proposed results in a short time with comparable BLEU scores.	BLEU Score – 0.13	Future work will be conducted to integrate this model with other pre-trained models to improve translation accuracy.

12.	Xin Ke.	2019	In this journal, the author implements a real-time synchronous translation method based on reinforcement learning.	The main limitation of this model is that it has to improve its performance regarding word embeddings and proper translation.	The experimental results show that the mixed real-time synchronous translation method and RL brings a certain degree of optimization.	ROGUE1- 36.71 ROGUE2- 15.74 ROGUE L- 36.22	In the future, Some effective pre-training models, such as CyberBERT, and inverse RL for better optimal results.
13.	Takatomo Kano, Sakriani Sakti, and Satoshi Nakamura.	2020	This journal proposes an attempt to build an end-to-end direct speech-to-text translation system on syntactically distant language pairs that suffer from long-distance reordering.	The main disadvantage of this model is that it focused only on a single language and may not work in multiple languages.	This model proposed results in a short time with high BLEU score when compared to other models.	BLEU Score – 0.34	Future work will be conducted to find effectiveness of proposed architecture to extend the application to various languages.
14.	Yoonseok Heo, Sangwoo Kang, Donghyun Hoo.	2019	The author proposed a multimodal neural machine translation system that uses both texts, and related images to translate Korean image captions into English.	The main limitation of this model is that it could not extract key feature from image for translation.	The proposed model improved the performance by +1.0 BLEU compared to the text-based NMT model.	BLEU Score- 0.30	Future work will extend the architecture by incorporating both visual and keyword components.
15.	Ovishake Sen, Mohtasim Fuad, MD. Nazrul Islam, Jakaria Rabba, Mehedi Masud, MD. Kamrul Hasan.	2022	The authors presented a thorough analysis of NLP and categorized it into 11 categories, Information Extraction, Machine Translation etc…		This journal presents a complete analysis of all the natural language processing methods which can be used to overcome future limitations.	Metrics used are ROGUE scores, F-Score, and Accuracy of different models.	Future work will be conducted to identify limitations and overcome them based upon the analysis of this journal.

16.	Chirantana Mallick, Ajit Kumar Das, Madhurima Dutta, Asit Kumar Das and Apurba Sarkar	2019	A graph-based text summarization method which captures the context of a text document. The method has been developed using modified TextRank.	The main limitation of this model is that multiple similar type of sentences with high score can be selected for the summary.	The proposed model outperforms other models for comparison in graph based summarization methods.	ROGUE-1 – 46.87.	Compare the method with many more summarization methods with different performance metrics.
17.	Hao Zeng and Guang Chen.	2020	The author proposed a model "lead3" of unsupervised extractive summarization and studied the relation between summarized sentences.	The main disadvantage of this model is that because of no training it takes more time and costs much more for summarizing.	The context information learned by this model can make the abstract better and the Diversity can affect the quality of the abstract.	ROGUE1- 40 ROGUE2- 17.	Future work will be conducted to identify position information since it is very important for summarization.
18.	Qianqian Xie, Jennifer Amy Bishop, Prayag Tiwari, Sophia Ananiadou.	2022	In this journal, the author has proposed KeBioSum, a novel knowledge infusion training framework for extractive summarization.	The main limitation of this model is that the PLMs have to be further enhanced for optimal results.	The proposed model outperforms strong baselines on the biomedical extractive summarization task.	ROGUE1- 42.9 ROGUE2- 37.	Further research will be conducted for abstractive summarization from this model and incorporate other language models.

19.	Aziz Qaroush, Ibrahim Abu Farha, Wasel Ghanem, Mahdi Washaha, Eman Maali	2021	In this journal, the author has proposed an automatic, generic, and extractive Arabic single document summarizing method.	The main disadvantage with this model is that it is necessary to identify the key features for extraction of key sentences.	This model proposed optimal results by extracting sentences based on their significance and importance with less redundancy.	Precision- 61.0, Recall- 67.0 and F1 Score- 64.0.	Future studies would investigate the methods to improve the presented approach by optimizing the weights of the extracted features
20.	Yuuki Iwasaki, Akihhiro Yamashita, Yoko Konno, Katsushi Matsubayashi.	2019	In this paper the author proposed an automatic abstractive text summarization algorithm in Japanese using a neural network.	The contents of the summary were repeated, the model was unable to handle unknown words, and there was a problem with simple word mistakes.	The model was able to learn correctly as the summary sentence captured the key points of the text to some extent.	Accuracy – 67%.	Future work will explore these limitations with new experiments and compare the results.
21.	Shehab Abdel-Salam and Ahmed Rafea.	2021	In this journal, the author proposed a BERT model for text summarization which is DistilBERT.	The main limitation of this model is that the context of the summary was not much accurate even with high ROGUE scores.	This model achieved the best results with a high accuracy of 98% and also with good ROGUE scores.	Accuracy - 98%.	Future work will be conducted on transforming this model to perform extractive summarization for specific use cases.

22.	Syed Abdul Basit Andrabi and Abdul Wahid.	2022	In this journal, the author proposed a neural network-based deep learning technique translation system for English to Urdu languages.	The main limitation of this model is that it could not translate all the words into the specified language and missed some words.	The translation quality of the proposed model was good the word error rates was also less.	BLEU Score – 0.45.	In the future, the aim is to increase the corpus size and include speech-to-text recognition using accurate models.
23.	Hritvik Gupta, Mayank Patel.	2021	In this journal, the author proposed a method of text summarization using LSA and Sentence based topic modeling with BERT.	The main limitation of this model is that it summarizes the text based on the words but not on the context of the summary.	The proposed model generates summaries based upon the semantics giving optimal results and performs accurately on large texts.	ROGUE1- 44.0 ROGUE L – 37.0	The future scope of this model is to generate more accurate summaries using abstractive summarizers.
24.	Anirudh Srikanth, Saravanan Thanu, Jaya Nirmala, Ashwin Shankar.	2020	The author brings an extractive summarization technique by using dynamic clustering and co-reference on BERT.	The main limitation of the model is that the entire document to be summarized could not be represented in a smaller number of sentences with proper context.	The proposed model gives us an optimal length of the summary with good ROGUE scores and accuracy without missing words.	ROGUE1- 41.4 ROGUE2- 17.9 ROGUE L- 37.9	The future work is to deploy the model on lectures from various online platforms like Coursera and Udemy.

25.	Kehai Chen, Tiejun Zhao, Muyun Yang, Lemao Liu, Akihiro Tamura, Rui Wang.	2019	The author proposed a novel neural approach to source dependence-based context representation for machine translation.	The main limitation of this model is that it has to improve its performance regarding word embeddings and proper translation.	The proposed model achieves significant improvement over the baseline systems and outperforms several existing context-enhanced methods.	BLEU Score- 0.17	The future scope of this model is to generate more models by integrating it with certain word embeddings.
26.	J.N. Madhuri, Ganesh Kumar. R.	2019	In this journal, the author proposed a novel statistical method to perform an extractive text summarization on a single document is demonstrated.	The main limitation is that it misses some words when converting the given summary into audio and it focuses only on sentence rankings but not on context.	Highly ranked sentences are extracted from the input document so it extracts important sentences that direct to a high-quality summary.	F1 Score – 62.29.	The future work is to deploy the model as an abstractive summarizer to produce much more meaningful summaries.
27.	Rohitash Chandra, and Venkatesh Kulkarni.	2022	In this journal, they have presented a framework that compares selected translations of the Bhagavad Gita using semantic and sentiment analyses on BERT.	The main limitation of this model is that it can only work on Sanskrit and could not produce high accurate translations when working on other languages.	The proposed model produced optimal translations without missing any words based on their importance without altering the real meaning.	Cosine Similarity Score- 62.0.	Future work focuses on using translators to review the sentiments presented in the different languages in different texts.

28.	Qianqian Xie, Jennifer Amy Bishop, Prayag Tiwari, Sophia Ananiadou.	2022	In this journal, the author has proposed KeBioSum, a novel knowledge infusion training framework for extractive summarization.	The main limitation of this model is that the PLMs have to be further enhanced for optimal results.	The proposed model outperforms strong baselines on the biomedical extractive summarization task.	ROGUE1- 42.9 ROGUE2- 37.	Further research will be conducted for abstractive summarization from this model and incorporate other language models.
29.	Mayank Ramina, Nihar Darnay, Chirag Ludbe, Ajay Dhruv.	2020	The author proposed an abstractive summarization model which uses advanced NLP capabilities of BERT to produce topic-level summary.	The main limitation of this model is that it misses most of the words due to which the chronological order gets missed in the produced summary.	The proposed model produces a meaningful topic - level summary without losing its context outperforming a few other abstractive models.	ROGUE1- 41.72 ROGUE2- 19.39 ROGUE L- 38.	The future research has to be conducted to provide much more context and improve the models using advanced NLP methods.
30.	Sunchit Sehgal, Badal Kumar, Maheshwar, Lakshay Rampal and Ankit Chaliya.	2019	In this paper, the author proposes Text Rank algorithm, to tackle the automatic summarization problem, and a variation to the similarity function.	The main limitation of this model is that there were a lot of duplicate sentences in the generated summary.	This journal is capable to extract a meaningful and coherent summary from the given input article.	Recall - 1.0. Precision - 0.49. F-score - 0.61.	Further research will be conducted to consider various other factors like personal pronouns and lexemes.

2.3 Heirarchy Diagram

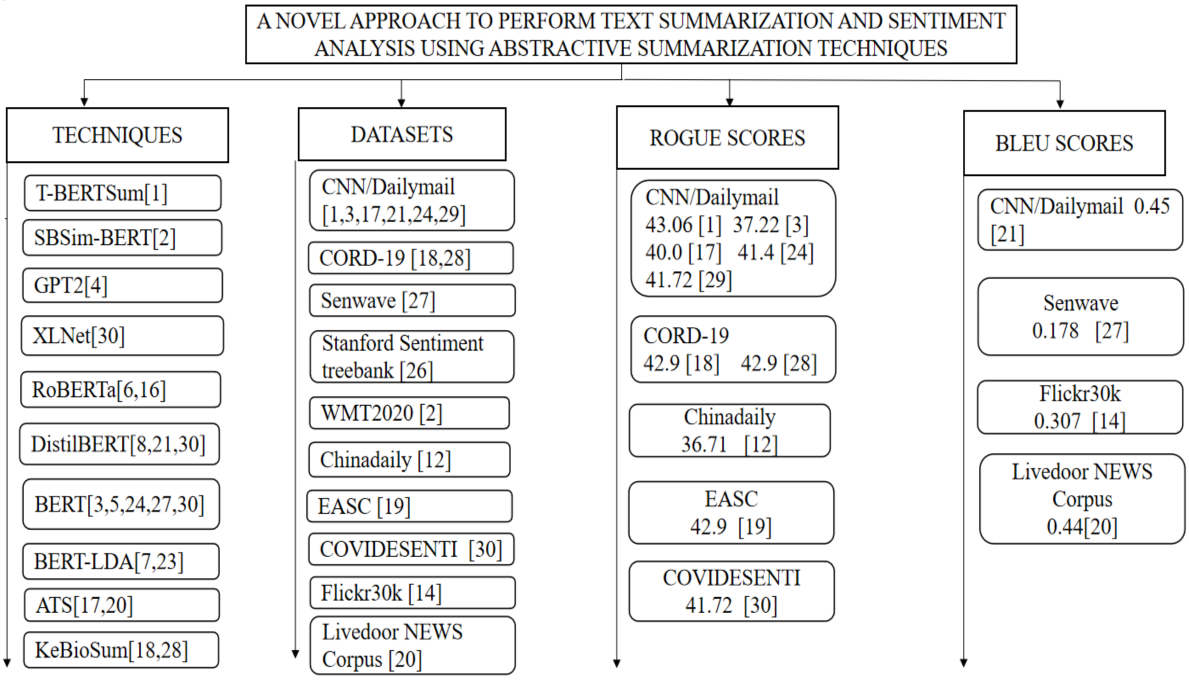

Fig 2.1 Heirarchy Diagram

Based on the analysis, the literature survey has given a complete structure of existing methodologies for whoever tries to summarise the data as illustrated in figure 2.1.

3. METHODOLOGY

This module explains the entire architecture of an hybrid model and also the preprocessing of the dataset and a brief explanation of all the models and methods that have been integrated to form this text summarization model for summarization of the data in a news article or a documentary This Module 3 explains all the components involved in workflow architecture with neat labeled figures.

3.1 Data Pre - Processing:

Preprocessing data is the first stage in the creation of any hybrid model. Our suggested hybrid strategy uses Pegasus, which has already received pre-training from more than 1.5 billion news stories and 350 million web pages. So, in order to do abstractive summarization, we do not require a significant amount of training data from our end. We used the XSum dataset, which contains 226,711 news stories and their summaries and is used for extreme summarization, for this project. Our results won't be accurate or efficient because the text data we used from the Xsum Dataset is in raw form and may contain several errors as well as undesired content. Pre-processing our data makes it easier to understand and is therefore required to achieve better results. The several steps in data pre-processing are as follows-

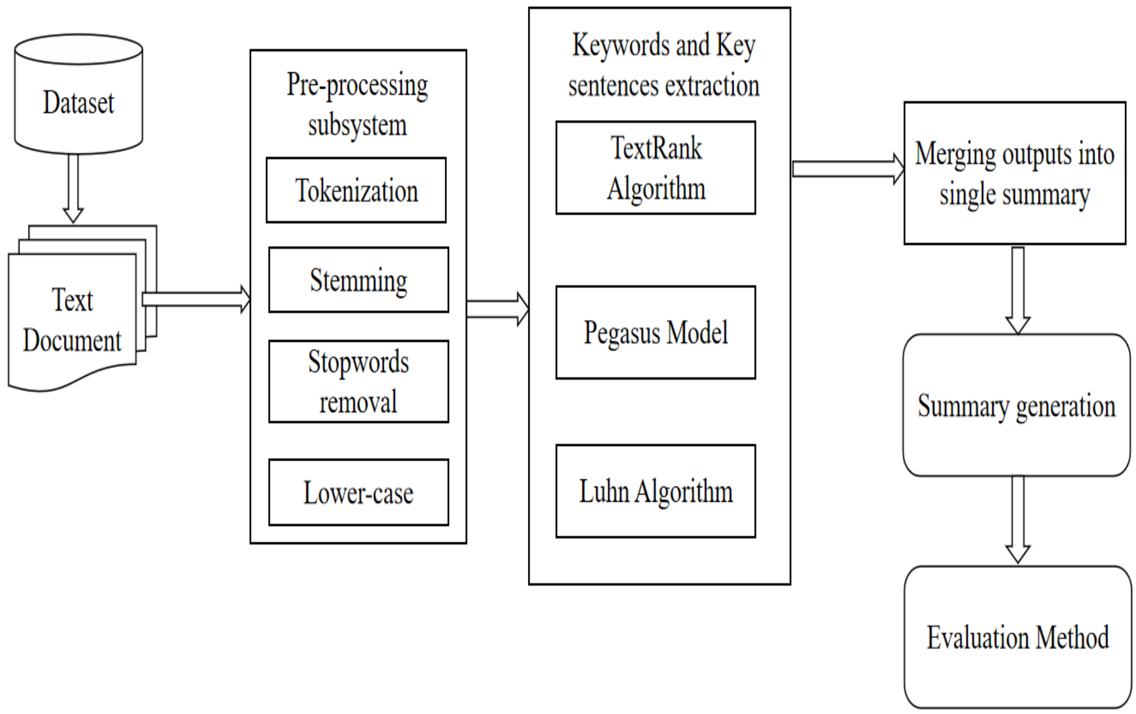

Fig 3.1.1 Framework of the hybrid model

The above figure 3.1.1 illustrates the framework of the proposed hybrid model. It gives a clear idea as to what are the algorithms and the steps involved in performing text summarization.

3.1.1 Tokenization:

In this step, we break down our text data into the smallest unit called tokens. Our dataset typically consists of a long text with numerous lines and lines made comprised of words. Long paragraphs are challenging to analyse, so we break them down into individual lines and then break those lines down into words. Tokens are the names for these words.

3.1.2 Stemming:

Stemming is a process used to remove any kind of suffix from a word and restore it to its root form , although often the root word produced by stemming is meaningless or does not belong in the English dictionary. Lemmatization, which creates a meaningful term after the suffix is removed, is an alternative to stemming.

3.1.3 Stop Words:

In any language, a stop word is a term that completes a sentence and gives it significance. for eg. Stop words in English include a variety of words like "I," "am," "are," "is to," etc. However, these stop-words are not very helpful for our model, so it is necessary to eliminate them from our dataset so that we may concentrate just on the relevant words and ignore the supporting words.

3.2 Proposed Hybrid Model and its workflow:

In this project, we have presented a hybrid model based on the Pegasus model, an abstractive summarization technique, and the Luhn and Textrank algorithms, which are extractive summarization techniques. The Luhn algorithm creates the final optimal paragraph by removing all the extraneous duplicate words and stopwords in the summarised paragraph after first using the text rank algorithm to rank the sentences according to their priority. This paragraph contains all the highly ranked sentences, so abstractive summarization is done on it using the Pegasus model to create a new summary with good context.

The algorithms and techniques used in this hybrid model are:

1. Text Rank Algorithm.
2. Pegasus.
3. Luhn Algorithm.

3.2.1 Textrank Algorithm:

The TextRank algorithm is the foundation of, a graph-based ranking model for text that identifies the most essential sentences in a document. Due to its unsupervised nature, TextRank is simple to utilize. [6,16,26,30] The algorithm creates a network with sentences as the nodes and overlapped words as the links after dividing the entire text into sentences. The most significant sentences in this network of sentences are identified by Page Rank.

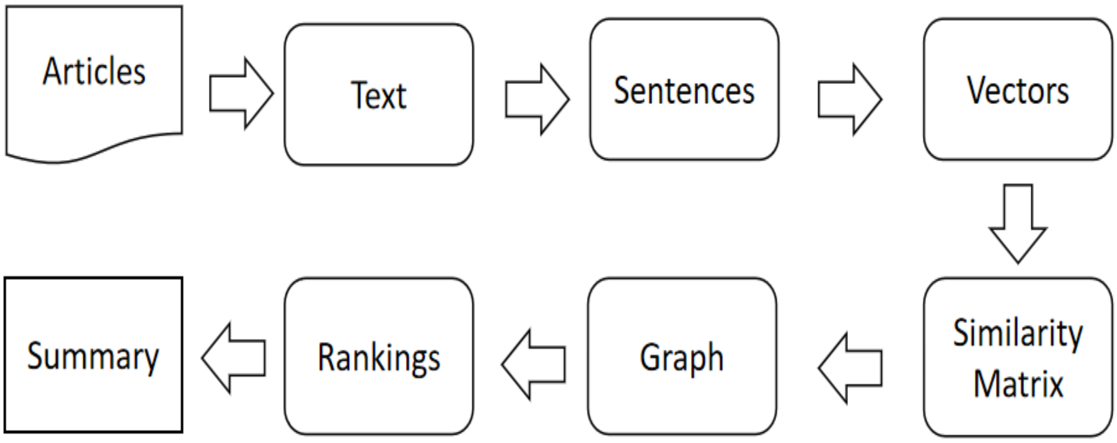

Fig 3.2.1. Framework of TextRank Algorithm.

By ranking each sentence in the text, Text Rank selects the most significant sentences. The top n sentences are used to construct a summary after sentences are ranked. A sentence's position in the resultant summary is unaffected by its rank. Instead, the original text's order of the chosen summary phrases is kept. The process begins with building a graph in which each node corresponds to a sentence from the source text that needs to be summarised. Then, we connect each sentence in this graph to additional phrases that are comparable. The edges of the resultant graph are these links. Each statement in this graph will point to other sentences that contain related information. The resulting edges of the graph are weighted. We then run a complex graph-based ranking formula over this weighted graph to determine the most important sentences in the original text and create the final summary. Let there be two sentences Si and Sj represented by a set of n words, let the words in Si be represented as Si = w i1, ..., w in. The similarity function for Si and Sj can be defined as shown in the following equation 1:

$$Sim(S_i, S_j) = \frac{|\{w_k | w_k \in S_i \ and \ w_k \in S_j\}|}{\log(|S_i| + \log(|S_j|))}$$ **Equation (1)**

The result of this process is a dense graph representing the document. From this graph, PageRank is used to compute the importance of each vertex. The most significant sentences are selected and presented in the same order as they appear in the document as the summary.

3.2.2 Pegasus Model:

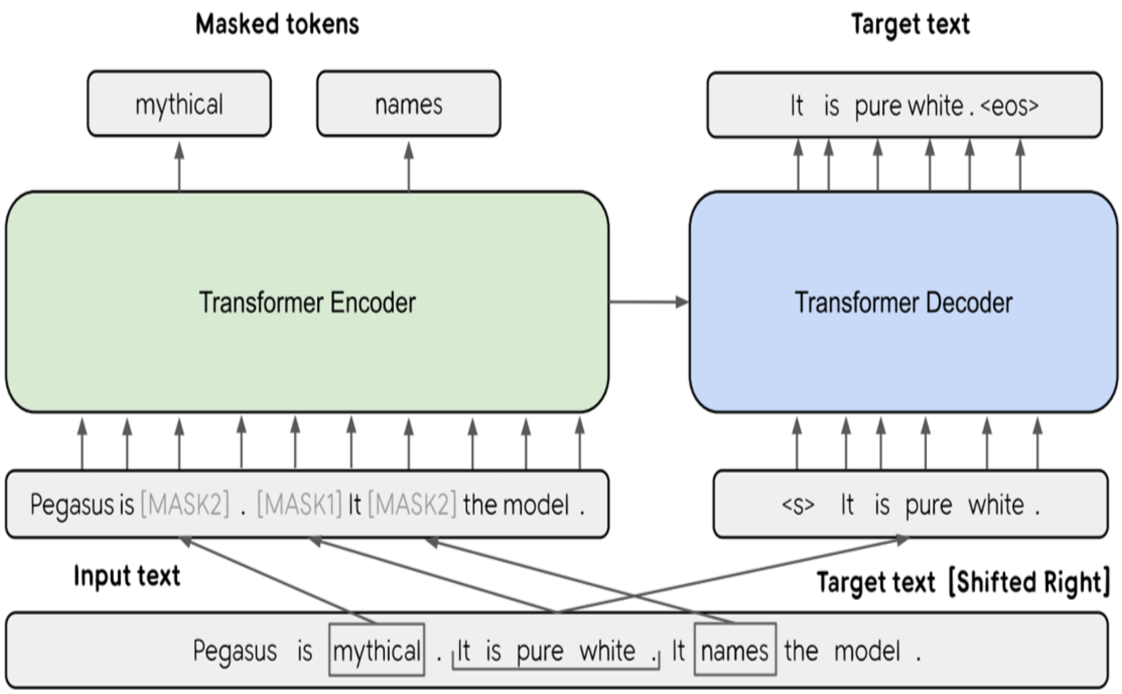

Figure 3.2.2 : Framework of Pegasus Model.

Pegasus stands for Pre-training with Extracted Gap sentences for Abstractive SUmmarization Pegasus use an encoder-decoder approach to learn sequences from sequences. According to this paradigm, the encoder will first examine the context of the entire input text before encoding it into a context vector, which is essentially a numerical representation of the input text. The decoder, whose task it is to decode the context vector and produce the summary, is then supplied this numerical representation. Pegasus employs a transformer-based encoder and decoder in place of conventional encoder-decoder architectures. Transformers are a set of systems designed to use a unique encoder-decoder architecture to convert an input sequence into an output sequence.

The special thing about transformers is the inclusion of a "self-attention" function and a few other modifications such as positional encoding. Text summaries created using transformers are usually of high quality and include original sentences.[5] PEGASUS is similar to other transformer models. The main differentiation is due to a unique method used during the model pre-training. The training text corpora's most significant sentences are hidden during PEGASUS pre-training. These sentences will be produced by the model as a single output sequence. It turns out that successful abstractive summarization requires a skill similar to the capacity to extract key lines from a text. The model has already been pre-trained on a huge number of news articles and web pages. On tiny datasets, the model can be improved further, and it performs wonderfully on text from a certain domain.

3.2.3 Luhn Algorithm :

Luhn Heuristic Method for text summarization is one of the earliest approaches to text summarization. Luhn’s algorithm is an approach based on TF-IDF. It selects only the words of higher importance as per their frequency. Higher weights are assigned to the words present at the beginning of the document. [30] Luhn’s method is a simple technique in order to generate a summary from given words.

The algorithm can be implemented in two stages

In the initial stage, we aim to determine which terms are more important to the document's meaning. According to Luhn, this is accomplished by first performing a frequency analysis and then identifying English terms that are significant but not crucial. The most frequent words in the document are identified in the second stage, and a selection of those that are less frequent but nonetheless significant in English is then selected. The following three steps are typically included in it:

1. It begins with transforming the content of sentences into a mathematical expression, or vector. Here we use a bag of words, which ignores all the filler words. Filler words are usually the supporting words that do not have any impact on our document's meaning. Then we count all the valuable words left to us. The words that are present at the beginning of the document are usually given much more importance and higher weights.

2. In this step, we evaluate sentences using the sentence scoring technique. We can use the scoring method as illustrated below in equation 2

$$Score = \frac{(\,Number\ of\ meaningful\ words\,)^2}{(\,Span\ of\ meaningful\ words)} \qquad \textbf{Equation (2)}$$

A span here refers to the part of the sentence/document consisting of all the meaningful words.

3. Once the sentence scoring is complete, the last step is simply to select those sentences with the highest overall rankings.

3.3 Evaluation Method:

In order to evaluate the summary that has been generated by the proposed hybrid model, we have used the ROUGE metric(Recall-Oriented Understudy for Gisting Evaluation) used for evaluating automatic summarization in natural language processing. [1,3-5] The ROGUE metrics are evaluated by comparing an automatically produced summary or translation against a reference or a set of references summary or translation. This method is discovered to be connected to human-generated precis because it is still reliant on Ngram data. Growing summaries don't have a single, perfect solution. Based on what is considered to be important statistics to cover, each precis produced by a human reader differs from another human reader. [12,16-18] We have also compared our proposed hybrid model with some advanced techniques such as BERT, GPT2, and XLNet to achieve higher ROGUE scores.

ROUGE-N: Overlap of n-grams between the system and reference summaries.

a) ROUGE-1 refers to the overlap of unigram (each word) between the system and reference summaries. [23-24]

b) ROUGE-2 refers to the overlap of bigrams which means two consecutive words between the system and reference summaries. [28-29]

We have deployed the above-mentioned hybrid model as a web application where the user can give any article as input in the HTML form and this input is stored in a variable which is then passed on for summarization using a proposed model where backend technologies have been used to connect the code implementation with the web application. After finetuning the model is saved in the back-end. This output summary is again displayed as an output on the HTML response page where the user can also perform text translation if needed based on the requirement. The web application has been styled with CSS and javascript. For translation purposes, we have used a basic translator package in python to perform the summary translation into the desired language.

3.4 Existing Models

Our model has been compared with three of the existing methodologies which are BERT, GPT2, and XLNet. Let’s understand all the ATS techniques.

3.4.1 BERT:

BERT stands for Bidirectional Encoder illustration from Transformers. It is designed to combine the left and right contexts to pre-teach deep bidirectional representations from unlabeled text. This enables us to improve our pre-trained BERT models by adding just one more output layer to produce models for a variety of recent NLP tasks. The Transformer building serves as the main headquarters for BERT. The entirety of Wikipedia (2.5 billion words!) and book corpora are among the large, fashionable unlabeled corpora on which BERT is pre-educated (800 million words). [1-3,27-29] The success of BERT is partially due to this pre-exercise step. this is because a model learns a deeper and more intimate understanding of how language functions when it is trained using a large text corpus. For almost any NLP challenge, this information will be available. A deep two-way model is BERT. At some point during the training phase, BERT learns information in a bidirectional manner from both the left and right side of the token's context.

3.4.2 GPT2:

Natural language processing (NLP) model GPT-2 is based on unsupervised machine learning methods. The syntactic, grammatical, and informational consistency of this framework allows it to finish and generate full parts of text. Models are capable of reading, comprehending, transcribing, summarising, and responding to enquiries about their structures and the data they hold. [4,8] The primary intention of GPT-2, a Transformer-based entirely language version, is to detect the next phrase in a sentence. Generative Pretrained Transformer 2 is open source and educated with over 1.5 billion parameters to generate the subsequent text order for the existing phrase. Appropriate textual content technology can be obtained for texts from distinctive domain names way to the range of datasets used in the education system.In comparison to GPT, GPT-2 has ten times as many parameters and ten times as many records. Without using the schooling records of specific domains, his GPT-2 from original texts can be used to determine his language obligations, including those of analysing, summarising, and translating.

3.4.3 XLNet:

XLNet is an autoregressive language model that uses a Transformer architecture with recursion to return joint distribution for a set of tokens. Its training goal is to determine word token probabilities that take into account all word token permutations in the set, not only the ones that are left or right of the target token. Modern BERT limitations are overcome by XLNet's generalised autoregressive pretraining technique, which maximises expected opportunity over all factored ordered permutations and permits bidirectional context today. [8] Additionally, XLNet includes the autoregressive Transformer-XL model into pre-education. Scientifically, in similar experimental conditions, XLNet surpasses his BERT on 20 tasks, including document rating, sentiment analysis, question answering, natural language reasoning, and many more.

4. RESULTS AND DISCUSSIONS

In the methodology section, a thorough explanation of the workflow was provided. The XSum dataset, which we used, initially had two properties or features, namely articles and their summaries, which were pre-processed by performing stemming, removing stop words, etc. We sequentially implemented each of the aforementioned abstraction approaches, with the output being fed into the inputs of the other techniques to create a final optimised summary.

In order to perform abstractive summarization, the cleaned and preprocessed paragraph was first fed into the text rank algorithm, which ranked the phrases according to their importance. The highest significant sentences were then supplied as input to the Pegasus model. After receiving this text as input, the Pegasus model performed abstractive summarization by rewriting the information and utilising new terms to make it appear as though it had been suggested by people. The Luhn method was then used to implement the summary paragraph in steps, producing the ultimate ideal paragraph as an output.

The pre-trained models BERT, GPT2, and XLNet were downloaded using the Hugging Face library. All of the experimentation was done in Google Colab. The system received the dataset Xsum, which was then immediately accessed using Google Colab instructions. This experiment was conducted in Python, and the Gensim package was used to implement the text rank algorithm. Hugging Face's transformers were used to create the Pegasus model, which was then adjusted on the XSum dataset. Utilizing the Sumy package, which includes multiple extractive summarization methods, the Luhn algorithm was put into practise.

We choose ROUGE, or remember-oriented Understudy for read the data Evaluation, as the evaluation metric in the text summary. In our proposed work, ROGUE-1 has been employed to evaluate and assess the current and proposed summaries. The articles that were summarised using the suggested hybrid model were additionally summarised using the BERT, GPT2, and XLNet models. Their outputs were saved in various variables, and all four output summaries were compared with the dataset's golden summary using the ROGUE scores. The average ROGUE scores for our model, which were based on summarization of various articles and their evaluation was roughly 56, which was also the highest among all the four models . The results of all the four models has been shown in the table 4.1 and also in figure 4.1 which is a bar graph comparing the results proposed by all the four models,

Table 4.1: Results Comparison Table

Article No.	**ROGUE-I SCORE**			
	Proposed Model	BERT	GPT2	XLNet
01.	56.0	54.5	45.0	46.0
02.	52.7	49.2	48.6	51.2
03.	47.6	48.5	51.0	45.9
04.	49.3	51.0	47.8	52.3
05.	54.2	49.5	48.3	54.5
06.	45.5	45.6	43.8	44.5
AVG.	51.0	49.6	47.4	49.0

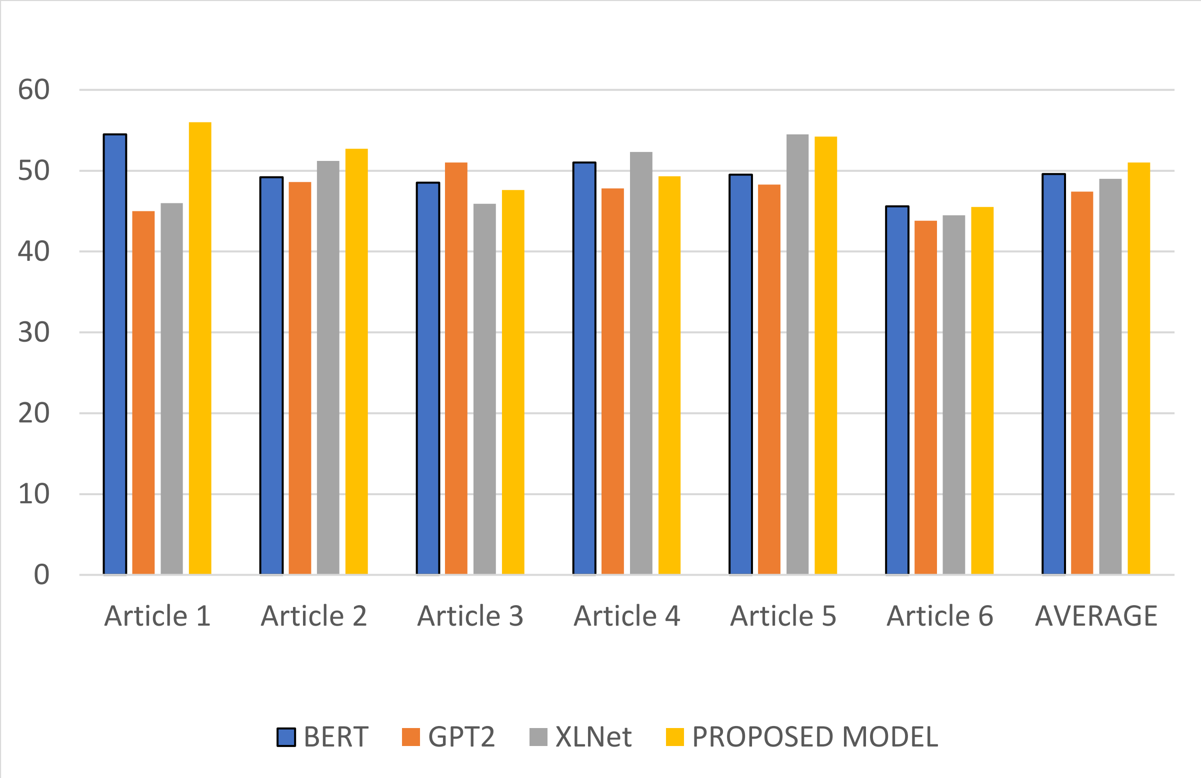

Figure 4.1: ATS Models vs proposed model

The aforementioned hybrid model has been implemented as an online application where users can input any article using an HTML form, store that input in a variable, and then send it on for summarizing. Backend technologies have been utilized to connect the code implementation and the web application. The model is saved at the back end after final adjustments. On the HTML response page, where the user can also execute text translation if necessary based on the requirement, this output summary is once more displayed as an output. Javascript and CSS have been used to style the online application. To conduct the summary translation into the required language, we have utilised a basic translator package in Python.

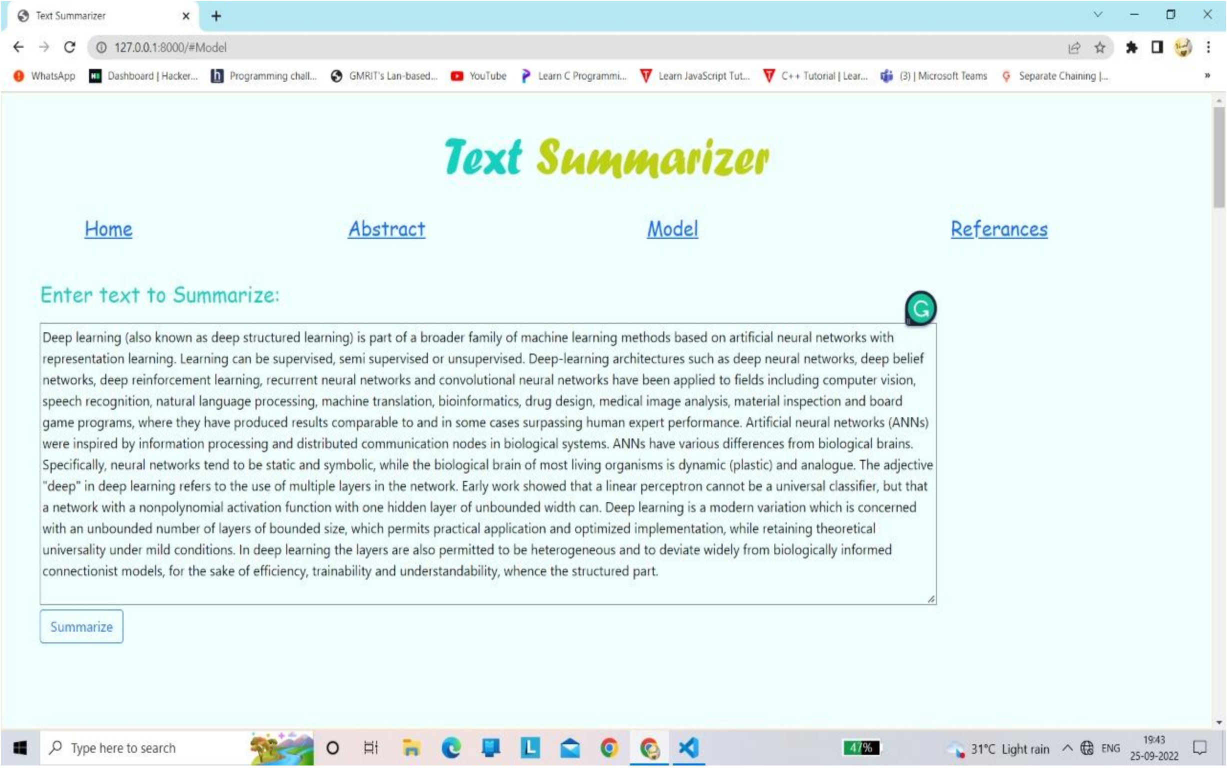

Fig 4.2: Web Deployment of proposed model

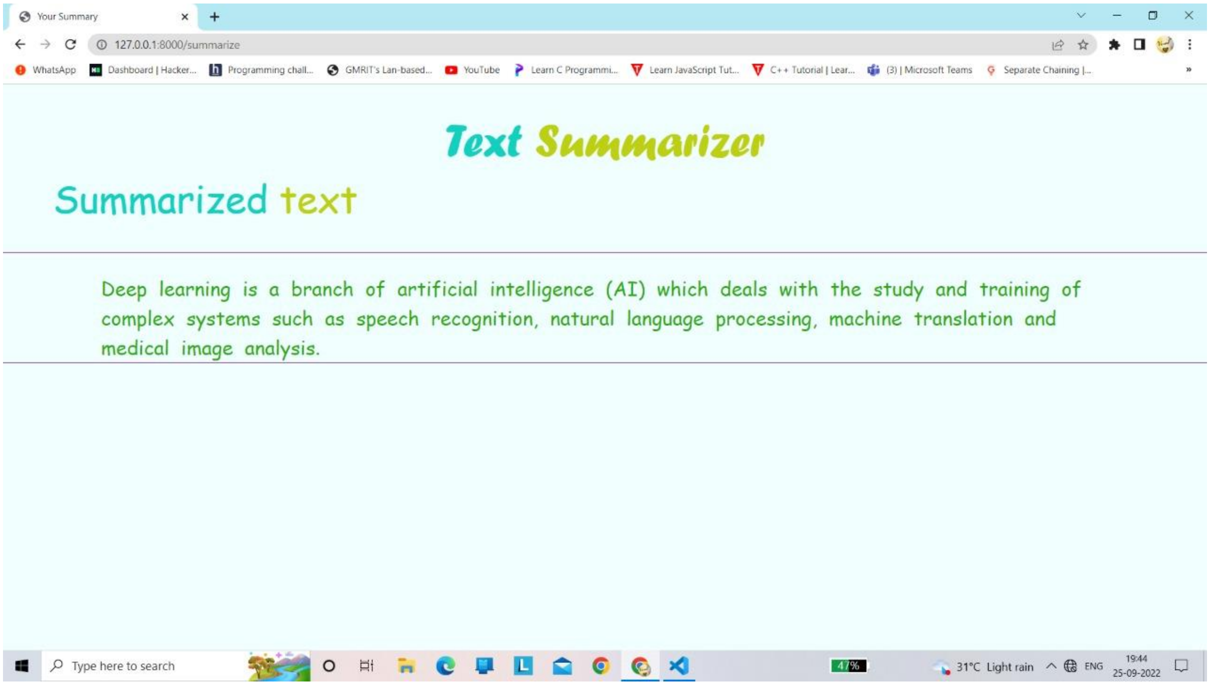

Fig 4.3: Web deployment summary

5. CONCLUSION AND FUTURE SCOPE

In this work, a hybrid model has been implemented which produced promising results as it achieved a higher average ROGUE-1 score when compared to BERT, GPT2, and XLNet. In this work, we have combined the extractive as well as abstractive summarization algorithms to take advantage of both the techniques, for this purpose we have used TextRank and Luhn extractive algorithms and Pegasus abstractive summarization technique. We have also studied various other automatic summarization techniques and compared them with our proposed model and we have found out that this type of hybridized technique produced better accurate results and also optimal paragraphs as summaries. A hybrid model was used in this study, and it showed promise because it outperformed BERT, GPT2, and XLNet in terms of average ROGUE-1 score. To benefit from both strategies, we merged extractive and abstractive summarization algorithms in this work. We did this by combining the TextRank and Luhn extractive algorithms with the Pegasus abstractive summarization technique. Our research into and comparison of a number of existing automatic summarising strategies with our suggested model revealed that the hybridised approach delivered more accurate findings and the best possible summaries.

Our main goal has been to improve our suggested hybrid model by giving it a final polish using the Xsum (extreme summarization) dataset. Additionally, we have made our work available as a web application and added translation to the paragraph that was summed up. By combining many different abstractive and extractive summarization techniques, we can further enhance the performance of this model. Additionally, we assessed the benefits of the suggested model compared to other models and carried out a quick analysis of a number of automatic summarising methods to comprehend their benefits, drawbacks, and methods of operation. Finally, we draw the conclusion that different merging algorithms or the integration of unsupervised summary approaches can be used to create hybrid summarization strategies. Additionally, there is a good chance that supervised and unsupervised summarization methods may be merged to create a far more potent summary model that could even deliver superior results.

The main goal of future work will be to improve the suggested hybrid model by further fusing it with certain unsupervised automatic summarization methods. This methodology can be used to summarise in numerous languages as well as just one. Future research will also be concentrated on using this hybrid approach to paragraphs that are extremely broad in addition to news and web articles. We can also alter our model's procedures to provide summaries that are more accurate and don't repeat or duplicate information. Even if efforts have been made to extract a lengthy and clear summary from the object, there is still a great deal of space for improvement in terms of how sentences are extracted and whether the summary follows its logical logic. Lexemes can also offer a comprehensive, logical, and logical summary of the item when several other elements, including personal pronouns, are taken into account.

APPENDIX

The implementation of the project is done with both the backend and frontend. The entire project is divided into five modules, they are preprocessing module, summarizing with the proposed module, summarizing with the existing module, summary translation module and finally the web-app implementation module.

The first four modules are done using natural language processing and Machine learning techniques. We encouraged Google Colab Notebook to implement the four modules. The last module is the web-app implementation module for which we created a web app to detect the weeds in the given input image.

After running the model in order to perform fine-tuning, the fine-tuned model i.e. the .h5 files will be generated in a specified “PATH”. We can integrate this model into the web-app and run the web-app to predict the weeds in a specific image uploaded.

MODULE-1:

```
!pip install spacy

import pandas as pd

import numpy as np

import spacy

from spacy.lang.en.stop_words import STOP_WORDS as stopwords

df=pd.read_csv('/content/drive/MyDrive/MAINPROJECT/XSum/train.csv',encoding='latin1')

df

df['article'].value_counts()

df['word_counts']=df['article'].apply(lambda x: len(str(x).split()))

df.sample(10)

df['word_counts'].max()

df['word_counts'].min()
```

```
def char_counts(x):
    s = x.split()
    x = ''.join(s)
    return len(x)
df['char_counts'] = df['article'].apply(lambda x: char_counts(str(x)))
df.sample(10)
df['avg_word_len'] = df['char_counts']/df['word_counts']
df.sample(10)
print(stopwords)
len(stopwords)
df['stop_words_len'] = df['article'].apply(lambda x: len([t for t in x.split() if t in stopwords]))
df.sample(10)
df[' hashtags_count'] = df['article' ] .apply(lambda x: len([t for t in x.split() if
t.startswith('@')]))
df['mention_count'] = df['article' ] .apply(lambda x: len([t for t in x.split() if
t.startswith('@')]))
df['numercis_count']=df['article'].apply(lambda x: len([ t for t in x.split() if t.isdigit()]))
df.sample(10)
df['article'] = df['article'].apply(lambda x: str(x).lower())
df.sample(10)
contractions = {
"ain't": "am not",
"aren't": "are not",
"car't": "cannot",
"can't've": "cannot have",
```

" cause": "because",

"could've": "could have",

"couldn't": "could not",

"couldn't've": "could not have",

"didn't": "did not",

"doesn't": "does not",

"don't": "do not",

"hadn't": "had not",

"hadn't've": "had not have",

"hasn't": "has not",

"haven't": "have not",

"he'd": "he would",

"he'd've": "he would have",

"he'll":"he will",

"he'll've": "he will have",

"he's": "he is",

"how'd": "how did",

"how'd'y": "how do you",

"how'll ": "how will",

"how's": "how does",

"i'd": "i would",

"i'd've" : "i would have",

"i'll": "i will",

"i'll've" : "i will have",

"i'm": "i am" ,

"i've": "i have",

"isn't": "is not",

"it'd": "it would",

"it'd've": "it would have",

"it'll": "it will",

"it'll've" : "it will have",

"it's": "it is",

"let's": "let us",

"ma'am": "madam",

"mayn't": "may not" ,

"might've" : "might have",

"mightn't": "might not",

"mightn't've": "might not have",

"must've": "must have",

"mustn't": "must not",

"mustn't've": "must not have",

"needn't": "need not",

"needn't've": "need not have",

"o'clock": "of the clock",

"oughtn't": "ought not",

"oughtn't've": "ought not have",

"shan't": "shall not",

"sha'n't": "shall not",

"she'd": "she would",

"she'd've": "she would have",

"she'll": "she will",

"she'll've": "she will have",

"she's": "she is",

"should've": "should have",

"shouldn't": "should not",

"shouldn It've": "should not have",

"so've": "so have",

"so's": "so is",

"that'd": "that would",

"that'd've": "that would have",

"that's": "that is",

"there'd" : "there would",

"there'd've": "there would have",

"there's": "there is",

"they'd": "they would",

"they'd've": "they would have",

"they'll": "they will",

"they'll've": "they will have",

"they're": "they are",

"they've": "they have",

"to've": "to have",

"wasn't": "was not",

```
" n ": " and ",
" u ": " you ",
" ur ": " your "}
def cont_to_exp(x):
    if type(x) is str:
        for key in contractions:
            value = contractions[key]
            x = x.replace(key, value)
        return x
    else:
        return x
df['article'] = df['article'].apply(lambda x: cont_to_exp(x))
import re
re.compile('<title>(.*)</title>')
df['article'] = df['article']. apply(lambda x: re.sub(r'[^\w ]+', "", x))
df.sample(10)
import unicodedata
def remove_accented_chars(x):
    x = unicodedata.normalize('NFKD' , x).encode(' ascii' , 'ignore').decode(' utf-8' , ' ignore')
    return x
df['article'] = df['article' ]. apply(lambda x: remove_accented_chars(x))
df['article_no_stop'] = df['article'].apply(lambda x: ' '.join([t for t in x.split() if t not in stopwords]))
df.sample(10)
```

example_text = """Deep learning (also known as deep structured learning) is part of a broader family of machine learning methods based on artificial neural networks with representation learning. Learning can be supervised, semi-supervised or unsupervised. Deep-learning architectures such as deep neural networks, deep belief networks, deep reinforcement learning, recurrent neural networks and convolutional neural networks have been applied to fields including computer vision, speech recognition, natural language processing, machine translation, bioinformatics, drug design, medical image analysis, material inspection and board game programs, where they have produced results comparable to and in some cases surpassing human expert performance. Artificial neural networks (ANNs) were inspired by information processing and distributed communication nodes in biological systems. ANNs have various differences from biological brains. Specifically, neural networks tend to be static and symbolic, while the biological brain of most living organisms is dynamic (plastic) and analogu."""

MODULE-2:

```
import gensim

from gensim.summarization import summarize

short_summary = summarize(example_text)

print(short_summary)

summary_by_ratio=summarize(example_text,ratio=0.1)

print(summary_by_ratio)

summary_by_word_count=summarize(example_text,word_count=60)

print(summary_by_word_count)

!pip install sentencepiece

!pip install transformers

!pip install torch torchvision torchaudio

!pip install datasets

from transformers import PegasusForConditionalGeneration, PegasusTokenizer, Trainer,
TrainingArguments
```

```
import torch
class PegasusDataset(torch.utils.data.Dataset):
    def __init__(self, encodings, labels):
        self.encodings = encodings
        self.labels = labels
    def __getitem__(self, idx):
        item = {key: torch.tensor(val[idx]) for key, val in self.encodings.items()}
        item['labels'] = torch.tensor(self.labels['input_ids'][idx])  # torch.tensor(self.labels[idx])
        return item
    def __len__(self):
        return len(self.labels['input_ids'])  # len(self.labels)
    def prepare_data(model_name,
                     train_texts, train_labels,
                     val_texts=None, val_labels=None,
                     test_texts=None, test_labels=None):
  tokenizer = PegasusTokenizer.from_pretrained(model_name)
  prepare_val = False if val_texts is None or val_labels is None else True
  prepare_test = False if test_texts is None or test_labels is None else True
  def tokenize_data(texts, labels):
    encodings = tokenizer(texts, truncation=True, padding=True)
    decodings = tokenizer(labels, truncation=True, padding=True)
    dataset_tokenized = PegasusDataset(encodings, decodings)
    return dataset_tokenized
  train_dataset = tokenize_data(train_texts, train_labels)
```

```
  val_dataset = tokenize_data(val_texts, val_labels) if prepare_val else None
  test_dataset = tokenize_data(test_texts, test_labels) if prepare_test else None
return train_dataset, val_dataset, test_dataset, tokenizer
def prepare_fine_tuning(model_name, tokenizer, train_dataset, val_dataset=None,
freeze_encoder=True, output_dir='./results'):
torch_device = 'cuda' if torch.cuda.is_available() else 'cpu'
model = PegasusForConditionalGeneration.from_pretrained(model_name).to(torch_device)
  if freeze_encoder:
    for param in model.model.encoder.parameters():
      param.requires_grad = False
   if val_dataset is not None:
    training_args = TrainingArguments(
      output_dir=output_dir,           # output directory
      num_train_epochs=1,           # total number of training epochs
      per_device_train_batch_size=1,   # batch size per device during training, can increase if
memory allows
      per_device_eval_batch_size=1,    # batch size for evaluation, can increase if memory
allows
      save_steps=500,                  # number of updates steps before checkpoint saves
      save_total_limit=5,              # limit the total amount of checkpoints and deletes the older
checkpoints
      evaluation_strategy='steps',     # evaluation strategy to adopt during training
      eval_steps=100,                  # number of update steps before evaluation
      warmup_steps=500,                # number of warmup steps for learning rate scheduler
      weight_decay=0.01,               # strength of weight decay
```

```
      logging_dir='./logs',            # directory for storing logs
      logging_steps=10,
    )
    trainer = Trainer(
      model=model,                         # the instantiated Hugging Transformers model to be trained
      args=training_args,                  # training arguments, defined above
      train_dataset=train_dataset,         # training dataset
      eval_dataset=val_dataset,            # evaluation dataset
      tokenizer=tokenizer
    )
  else:
    training_args = TrainingArguments(
      output_dir=output_dir,           # output directory
      num_train_epochs=1,              # total number of training epochs
      per_device_train_batch_size=1,   # batch size per device during training, can increase if memory allows
      save_steps=500,                  # number of updates steps before checkpoint saves
      save_total_limit=5,              # limit the total amount of checkpoints and deletes the older checkpoints
      warmup_steps=500,                # number of warmup steps for learning rate scheduler
      weight_decay=0.01,               # strength of weight decay
      logging_dir='./logs',            # directory for storing logs
      logging_steps=10,
    )
```

```
    trainer = Trainer(
      model=model,                         # the instantiated hugging Transformers model to be trained
      args=training_args,                  # training arguments, defined above
      train_dataset=train_dataset,         # training dataset
      tokenizer=tokenizer
    )
  return trainer
if __name__=='__main__':
  # use XSum dataset as example, with first 1000 docs as training data
  from datasets import load_dataset
  dataset = load_dataset("xsum")
  train_texts, train_labels = dataset['train']['document'][:50], dataset['train']['summary'][:50]
  model_name = 'google/pegasus-large'
  train_dataset, _, _, tokenizer = prepare_data(model_name, train_texts, train_labels)
  trainer = prepare_fine_tuning(model_name, tokenizer, train_dataset)
  trainer.train()
from transformers import PegasusForConditionalGeneration
from transformers import PegasusTokenizer
from transformers import pipeline
model_name = "google/pegasus-xsum"
pegasus_tokenizer = PegasusTokenizer.from_pretrained(model_name)
example_text = """Deep learning (also known as deep structured learning) is part of a broader family of machine learning methods based on artificial neural networks with representation learning. Learning can be supervised, semi-supervised or unsupervised. Deep-learning architectures such as deep neural networks, deep belief networks, deep
```

reinforcement learning, recurrent neural networks and convolutional neural network have been applied to fields including computer vision, speech recognition, natural language processing, machine translation, bioinformatics, drug design, medical image analysis, material inspection and board game programs, where they have produced results comparable to and in some cases surpassing human expert performance. Artificial neural networks (ANNs) were inspired by information processing and distributed communication nodes in biological systems. ANNs have various differences from biological brains. Specifically, neural networks tend to be static and symbolic, while the biological brain of most living organisms is dynamic (plastic) and analogue."""

```
pegasus_model = PegasusForConditionalGeneration.from_pretrained(model_name)

tokens = pegasus_tokenizer(example_text, truncation=True, padding="longest",
return_tensors="pt")

# Summarize text

encoded_summary = pegasus_model.generate(**tokens)

# Decode summarized text

decoded_summary = pegasus_tokenizer.decode(

      encoded_summary[0],

      skip_special_tokens=True

)

print(decoded_summary)

summarizer = pipeline(

    "summarization",

    model=model_name,

    tokenizer=pegasus_tokenizer,

    framework="pt"

)

summary = summarizer(example_text, min_length=30, max_length=150)
```

```
summary[0]["summary_text"]

!pip install sumy

import sumy

import nltk

nltk.download('punkt')

from sumy.summarizers.luhn import LuhnSummarizer

from sumy.nlp.tokenizers import Tokenizer

from sumy.parsers.plaintext import PlaintextParser

parser=PlaintextParser.from_string(example_text,Tokenizer('english'))

luhn_summarizer=LuhnSummarizer()

luhn_summary=luhn_summarizer(parser.document,sentences_count=3)

for sentence in luhn_summary:

  print(sentence)
```

MODULE-3:

```
"""USING BERT"""

!pip install transformers==4.5.1

!pip install bert-extractive-summarizer

!pip install spacy==2.0.12

pip uninstall thinc

pip uninstall cymem

pip install spacy

from summarizer import Summarizer,TransformerSummarizer

bert_model = Summarizer()

bert_summary = ''.join(bert_model(example_text, min_length=60))
```

```
print(bert_summary)

"""# Text Summarization using GPT-2"""

GPT2_model =
TransformerSummarizer(transformer_type="GPT2",transformer_model_key="gpt2-
medium")

full = ''.join(GPT2_model(example_text, min_length=60))

print(full)

"""# Text Summarization using XLNET"""

model =
TransformerSummarizer(transformer_type="XLNet",transformer_model_key="xlnet-base-
cased")

full = ''.join(model(example_text, min_length=60))

print(full)
```

MODULE-4:

```
"""# SUMMARIZED TEXT TRANSLATION"""

text_sample="Deep learning (also known as deep structured learning) is part of a broader family of machine learning methods based on artificial neural networks with representation learning. In deep learning the layers are also permitted to be heterogeneous and to deviate widely from biologically informed connectionist models, for the sake of efficiency, trainability and understandability."

import re

wordList = re.sub("[^\w]", " ",  text_sample).split()

print(wordList)

text_one=' '.join(wordList)

print(text_one)

!pip install googletrans
```

```
from googletrans import Translator

translator = Translator()

print(translator.detect(text_one))

output= translator.translate(text_one, src='en',dest='telugu')

print(output.text)

pip install -r rouge/requirements.txt

pip install rouge-score

from rouge_score import rouge_scorer

scorer = rouge_scorer.RougeScorer(['rouge1'], use_stemmer=True)

from rouge_score import rouge_scorer

scorer = rouge_scorer.RougeScorer(['rouge1'], use_stemmer=True)

scores = scorer.score('Deep learning is a family of machine learning methods. Learning can be supervised, semi-supervised or unsupervised. Deep-learning architectures have been applied to fields including computer vision, speech recognition, natural language processing and bioinformatics.' , 'Deep learning is part of a broader family of machine learning methods based on artificial neural networks with representation learning. Deep-learning architectures such as deep neural networks, deep reinforcement learning, recurrent neural networks and convolutional neural networks have been applied to fields including computer vision, speech recognition, natural language processing, machine translation, bioinformatics, drug design, medical image analysis, material inspection and board game programs.')

print(scores)
```

MODULE-5:

```
import os
from unittest import result
from flask import Flask, render_template,request

from transformers import PegasusForConditionalGeneration
from transformers import PegasusTokenizer
```

```
from transformers import pipeline
model_name = "google/pegasus-xsum"
pegasus_tokenizer = PegasusTokenizer.from_pretrained(model_name)
pegasus_model = PegasusForConditionalGeneration.from_pretrained(model_name)

app = Flask(__name__)
picFolderr=os.path.join('static','pics')
app.config['UPLOAD_FOLDER']=picFolderr
@app.route("/")

def msg():
    pic1=os.path.join(app.config['UPLOAD_FOLDER'],'model.jpg')
    return render_template("index.html",model_img=pic1)
@app.route("/summarize", methods=['POST','GET'])
def getSummary():
    body=request.form['data']
    text=body.split()
    cnt=0
    for i in text:
        cnt+=1
    summarizer = pipeline(
        "summarization",
        model=model_name,
        tokenizer=pegasus_tokenizer,
        framework="pt"
    )
    summary = summarizer(body, min_length=80, max_length=450)
    result=summary[0]["summary_text"]
    return render_template('summary.html',result=result)

if __name__ =="__main__":
    app.run(debug=True,port=8000)

from unittest import result
```

```
from flask import Flask, render_template,request
app = Flask(__name__)
@app.route("/")

def msg():
    return render_template("index.html")
@app.route("/summarize", methods=['POST','GET'])
def output():
    result=res
    return render_template('summary.html',result=result)

if __name__ =="__main__":
    app.run(debug=True,port=8800)
```

REFERENCES

[1] Ma, T., Pan, Q., Rong, H., Qian, Y., Tian, Y., & Al-Nabhan, N. (2021). T-bertsum: Topic-aware text summarization based on bert. *IEEE Transactions on Computational Social Systems*, *9*(3), 879-890.

[2] Mrinalini, K., Vijayalakshmi, P., & Nagarajan, T. (2022). SBSim: A Sentence-BERT Similarity-Based Evaluation Metric for Indian Language Neural Machine Translation Systems. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, *30*, 1396-1406.

[3] Wang, Q., Liu, P., Zhu, Z., Yin, H., Zhang, Q., & Zhang, L. (2019). A text abstraction summary model based on BERT word embedding and reinforcement learning. *Applied Sciences*, *9*(21), 4701.

[4] Li, P., Yu, J., Chen, J., & Guo, B. (2021). HG-News: News Headline Generation Based on a Generative Pre-Training Model. *IEEE Access*, *9*, 110039-110046.

[5] Gidiotis, A., & Tsoumakas, G. (2020). A divide-and-conquer approach to the summarization of long documents. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, *28*, 3029-3040.

[6] Akhtar, N., Beg, M. S., & Javed, H. (2019, August). TextRank enhanced topic model for query focussed text summarization. In *2019 Twelfth International Conference on Contemporary Computing (IC3)* (pp. 1-6). IEEE.

[7] Tan, X., Zhuang, M., Lu, X., & Mao, T. (2021). An analysis of the emotional evolution of large-scale internet public opinion events based on the BERT-LDA hybrid model. *IEEE Access*, *9*, 15860-15871.

[8] Mridha, M. F., Lima, A. A., Nur, K., Das, S. C., Hasan, M., & Kabir, M. M. (2021). A survey of automatic text summarization: Progress, process and challenges. *IEEE Access*, *9*, 156043-156070.

[9] Vathsala, M. K., & Holi, G. (2020). RNN based machine translation and transliteration for Twitter data. *International Journal of Speech Technology*, *23*(3), 499-504.

[10] Bawa, S., & Kumar, M. (2021). A comprehensive survey on machine translation for English, Hindi and Sanskrit languages. *Journal of Ambient Intelligence and Humanized Computing*, 1-34.

[11] Ning, J., & Ban, H. (2021). Design and Testing of Automatic Machine Translation System Based on Chinese-English Phrase Translation. *Mobile Information Systems*, *2021*.

[12] Ke, X. (2022). English synchronous real-time translation method based on reinforcement learning. *Wireless Networks*, 1-13.

[13] Kano, T., Sakti, S., & Nakamura, S. (2020). End-to-end speech translation with transcoding by multi-task learning for distant language pairs. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, *28*, 1342-1355.

[14] Heo, Y., Kang, S., & Yoo, D. (2019). Multimodal neural machine translation with weakly labeled images. *IEEE Access*, *7*, 54042-54053.

[15] Sen, O., Fuad, M., Islam, M. N., Rabbi, J., Masud, M., Hasan, M. K., ... & Iftee, M. A. R. (2022). Bangla Natural Language Processing: A Comprehensive Analysis of Classical, Machine Learning, and Deep Learning Based Methods. *IEEE Access*.

[16] Mallick, C., Das, A. K., Dutta, M., Das, A. K., & Sarkar, A. (2019). Graph-based text summarization using modified TextRank. In *Soft computing in data analytics* (pp. 137-146). Springer, Singapore.

[17] Zeng, H., & Chen, G. (2020, December). Unsupervised extractive summarization based on context information. In *2020 IEEE 6th International Conference on Computer and Communications (ICCC)* (pp. 1651-1655). IEEE.

[18] Xie, Q., Bishop, J. A., Tiwari, P., & Ananiadou, S. (2022). Pre-trained language models with domain knowledge for biomedical extractive summarization. *Knowledge-Based Systems*, 109460.

[19] Qaroush, A., Farha, I. A., Ghanem, W., Washaha, M., & Maali, E. (2021). An efficient single document Arabic text summarization using a combination of statistical and semantic features. *Journal of King Saud University-Computer and Information Sciences*, *33*(6), 677-692.

[20] Iwasaki, Y., Yamashita, A., Konno, Y., & Matsubayashi, K. (2019, November). Japanese abstractive text summarization using BERT. In *2019 International Conference on Technologies and Applications of Artificial Intelligence (TAAI)* (pp. 1-5). IEEE

[21] Abdel-Salam, S., & Rafea, A. (2022). Performance Study on Extractive Text Summarization Using BERT Models. *Information*, *13*(2), 67.

[22] Andrabi, S. A. B., & Wahid, A. (2022). Machine translation system using deep learning for English to Urdu. *Computational Intelligence and Neuroscience*, *2022*.

[23] Gupta, H., & Patel, M. (2021, March). Method Of Text Summarization Using Lsa And Sentence Based Topic Modelling With Bert. In *2021 International Conference on Artificial Intelligence and Smart Systems (ICAIS)* (pp. 511-517). IEEE.

[24] Srikanth, A., Umasankar, A. S., Thanu, S., & Nirmala, S. J. (2020, October). Extractive text summarization using dynamic clustering and co-reference on BERT. In *2020 5th International Conference on Computing, Communication and Security (ICCCS)* (pp. 1-5). IEEE.

[25] Chen, K., Zhao, T., Yang, M., Liu, L., Tamura, A., Wang, R., ... & Sumita, E. (2019). A neural approach to source dependence based context model for statistical machine translation. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, *26*(2), 266-280.

[26] Madhuri, J. N., & Kumar, R. G. (2019, March). Extractive text summarization using sentence ranking. In *2019 International Conference on Data Science and Communication (IconDSC)* (pp. 1-3). IEEE.

[27] Chandra, R., & Kulkarni, V. (2022). Semantic and sentiment analysis of selected bhagavad gita translations using BERT-based language framework. *IEEE Access*, *10*, 21291-21315.

[28] Xie, Q., Bishop, J. A., Tiwari, P., & Ananiadou, S. (2022). Pre-trained language models with domain knowledge for biomedical extractive summarization. *Knowledge-Based Systems*, 109460.

[29] Ramina, M., Darnay, N., Ludbe, C., & Dhruv, A. (2020, May). Topic level summary generation using BERT induced Abstractive Summarization Model. In *2020 4th International Conference on Intelligent Computing and Control Systems (ICICCS)* (pp. 747-752). IEEE.

[30] Sehgal, S., Kumar, B., Rampal, L., & Chaliya, A. (2019). A modification to graph based approach for extraction based automatic text summarization. In *Progress in advanced computing and intelligent engineering* (pp. 373-378). Springer, Singapore.

LIST OF PUBLICATIONS

[1] Sai Kiran, Vinod Kumar, Tulasi Ram, Venkata Ramana, Surendra, Srividya. (2022). A Review Of Extractive And Abstractive Text Summarization Techniques. In *International Journal of Advances in Engineering and Management (IJAEM)* (pp.1020-1027).