ABSTRACTIVE TEXT SUMMARIZATION USING BART MODEL

A main Project Report submitted in the partial fulfillment of the requirements for the award of the degree

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE AND ENGINEERING

Submitted by

P.Gayathri	**(19471A05N6)**
M.Naga Sirisha	**(20475A0505)**
CH.Dhanalakshmi	**(19471A05K3)**

Under the esteemed guidance of

Sd.Rizwana **MTech.,(Ph.D)**

Assistant Professor

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

NARASARAOPETA ENGINEERING COLLEGE NARASARAOPET (AUTONOMOUS)

Accredited by NAAC with A+ Grade and NBA under Cycle-1

NIRF rank in the band of 251-320 and an ISO 9001:2015 Certified

Approved by AICTE , New Delhi, Permanently Affiliated to JNTUK, Kakinada

KOTAPPAKONDA ROAD, YALAMANDA VILLAGE, NARASARAOPET-522601

2022-2023

NARASARAOPETA ENGINEERING COLLEGE: NARASARAOPETA

(AUTONOMOUS)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

This is to certify that the main project entitled **"ABSTRACTIVE TEXT SUMMARIZATION"** is a bonafide work done by **P. Gayathri(19471A05N6), Ch. Dhana Lakshmi (19471A05K3), M. Naga Sirisha(20475A0505)** in partial fulfilment of the requirements for the award of the degree of **BACHELOR OF TECHNOLOGY** in the department of **COMPUTER SCIENCE AND ENGINEERING during 2022-2023.**

Mrs. Sd. Rizwana **MTech.,(Ph.D.)**
PROJECT GUIDE

Mrs. M. Sireesha ,MTech.,(Ph.D)
PROJECT CO-ORDINATOR

Dr.S.N. TirumalaRao **M.Tech.,Ph.D**
HEAD OF THE DEPARTMENT

EXTERNAL EXAMINER

ACKNOWLEDGEMENT

We wish to express our thanks to carious personalities who are responsible for the completion of the project. We are extremely thankful toour beloved chairman sir **M.V. Koteswara Rao**, **B.sc** who took keen interest in us in every effort throughout this course. We owe out gratitude to our principal **Dr.M.Sreenivasa Kumar, M.Tech,Ph.D(UK),MISTE,FIE(1)** for his kind attention and valuable guidance throughout the course.

We express our deep felt gratitude to **Dr.S.N.Tirumala Rao,M.Tech, Ph.D** H.O.D. CSE department and our guide **Sd.Rizwana,M.Tech.,(Ph.D)** Assistant Professor of CSE department whose valuable guidance and unstinting encouragement enable us to accomplish our project successfully in time.

We extend our sincere thanks to **Dr.M.Sireesha,M.Tech,(Ph.D)** Associate Professor and coordinator of the project for extending her encouragement. Their profound knowledge and willingness have been a constant source of inspiration for us throughout this project work.

We extend our sincere thanks to all other teaching and non-teaching staff to department for their cooperation and encouragement during our B.Tech degree. We have no words to acknowledge the warm affection, constant inspiration and encouragement that we receive from our parents.

We affectionately acknowledge the encouragement received from our friends those who involved in giving valuable suggestions had clarifying out all doubts which had really helped us in successfully completing our project.

By

P.Gayathri	**(19471A05N6)**
M.Naga Sirisha	**(20475A0505)**
Ch.Dhanalakshmi	**(19471A05K3)**

ABSTRACT

A new abstractive summarization model for documents, hierarchical BART (Hie-BART), which captures the hierarchical structures of documents (i.e., their sentence-word structures) in the BART model. Although the existing BART model has achieved state-of-the-art performance on document summarization tasks, it does not account for interactions between sentence-level and word-level information. In machine translation tasks, the performance of neural machine translation models can be improved with the incorporation of multi-granularity self-attention (MG-SA), which captures relationships between words and phrases. Inspired by previous work, the proposed HieBART model incorporates MG-SA into the encoder of the BART model for capturing sentence-word structures. Evaluations performed on the CNN/Daily Mail dataset show that the proposed Hie-BART model outperforms strong baselines and improves the performance of a non-hierarchical BART model.

NARASARAOPETA
NEC ENGINEERING COLLEGE
(AUTONOMOUS)

INSTITUTE VISION AND MISSION

INSTITUTION VISION

To emerge as a Centre of excellence in technical education with a blend of effective student centric teaching learning practices as well as research for the transformation of lives and community.

INSTITUTION MISSION

M1: Provide the best class infra-structure to explore the field of engineering and research.

M2: Build a passionate and a determined team of faculty with student centric teaching, imbibing experiential, innovative skills.

M3: Imbibe lifelong learning skills, entrepreneurial skills and ethical values in students for addressing societal problems.

NARASARAOPETA ENGINEERING COLLEGE
(AUTONOMOUS)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

VISION OF THE DEPARTMENT

To become a center of excellence in nurturing the quality Computer Science & Engineering professionals embedded with software knowledge, aptitude for research and ethical values to cater to the needs of industry and society.

MISSION OF THE DEPARTMENT

The department of Computer Science and Engineering is committed to

M1: Mould the students to become Software Professionals, Researchers and Entrepreneurs by providing advanced laboratories.

M2: Impart high quality professional training to get expertise in modern software tools and technologies to cater to the real time requirements of the Industry.

M3: Inculcate team work and lifelong learning among students with a sense of societal and ethical responsibilities.

NEC NARASARAOPETA ENGINEERING COLLEGE (AUTONOMOUS)

Program Specific Outcomes (PSO's)

PSO1: Apply mathematical and scientific skills in numerous areas of Computer Science and Engineering to design and develop software-based systems.

PSO2: Acquaint module knowledge on emerging trends of the modern era in Computer Science and Engineering

PSO3: Promote novel applications that meet the needs of entrepreneur, environmental and social issues.

NARASARAOPETA ENGINEERING COLLEGE

(AUTONOMOUS)

Program Educational Objectives (PEO's)

The graduates of the programme are able to:

PEO1: Apply the knowledge of Mathematics, Science and Engineering fundamentals to identify and solve Computer Science and Engineering problems.

PEO2: Use various software tools and technologies to solve problems related to academia, industry and society.

PEO3: Work with ethical and moral values in the multi-disciplinary teams and can communicate effectively among team members with continuous learning.

PEO4: Pursue higher studies and develop their career in software industry.

NARASARAOPETA
NEC ENGINEERING COLLEGE
(AUTONOMOUS)

Program Outcomes

1. **Engineering knowledge:** Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
2. **Problem analysis:** Identify, formulate, research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.
3. **Design/development of solutions:** Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.
4. **Conduct investigations of complex problems:** Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.
5. **Modern tool usage:** Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modelling to complex engineering activities with an understanding of the limitations.
6. **The engineer and society:** Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.

7. **Environment and sustainability:** Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and needfor sustainable development.

8. **Ethics:** Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.

9. **Individual and team work**: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.

10. **Communication**: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.

11. **Project management and finance**: Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.

12. **Life-long learning**: Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

NEC NARASARAOPETA ENGINEERING COLLEGE
(AUTONOMOUS)

Project Course Outcomes (CO'S):

CO425.1: Analyse the System of Examinations and identify the problem.

CO425.2: Identify and classify the requirements.

CO425.3: Review the Related Literature**. CO425.4:** Design and Modularize the project.

CO425.5: Construct, Integrate, Test and Implement the Project.

CO425.6: Prepare the project Documentation and present the Report using appropriate method.

Course Outcomes – Program Outcomes mapping

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
C425.1		✓											✓		
C425.2	✓		✓		✓								✓		
C425.3				✓		✓	✓	✓					✓		
C425.4			✓			✓	✓	✓					✓	✓	
C425.5					✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
C425.6									✓	✓	✓		✓	✓	

Course Outcomes – Program Outcome correlation

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
C425.1	2	3											2		
C425.2			2		3								2		
C425.3				2		2	3	3					2		
C425.4			2			1	1	2					3	2	
C425.5					3	3	3	2	3	2	2	1	3	2	1
C425.6									3	2	1		2	3	

Note: The values in the above table represent the level of correlation between CO’s and PO’s:

1. Low level
2. Medium level
3. High level

Project mapping with various courses of Curriculum with Attained PO’s:

Name of the course from which principles are applied in this project	**Description of the device**	**Attained PO**
C3.2.4, C3.2.5	Gathering the requirements and defining the problem, plan to develop a smart bottlefor health care using sensors.	PO1, PO3
CC4.2.5	Each and every requirement is critically analyzed, the process model is identified and divided into five modules	PO2, PO3
CC4.2.5	Logical design is done by using the unified modelling language which involves individual team work	PO3, PO5, PO9
CC4.2.5	Each and every module is tested,integrated, and evaluated in our project	PO1, PO5
CC4.2.5	cumentation is done by all our four members inthe form of a group	PO10
CC4.2.5	ch and every phase of the work ingroup is presented periodically	PO10, PO11
CC4.2.5	Implementation is done and the project will be handled by the hospital management and infuture updates in our project can be done based on airbubbles occurring in liquid in saline.	PO4, PO7
CC4.2.8 CC4.2.	The physical design includes hardware components like sensors, gsm module,software and Arduino.	PO5, PO6

INDEX

List of figures

1. INTRODUCTION

In recent years, improvements to abstractive document summarization models have been developed through the incorporation of pre-training. The BERTSUM model (Liu and Lapata, 2019) has been proposed as a pre-training model for document summarization tasks. For sequence-to-sequence tasks, the T5 model (Raffel et al., 2020) and the BART model (Lewis et al., 2020) have been proposed as part of generalized pre-training models. Among the existing pre-training models, the BART model achieves state-of-the-art performance on document summarization tasks. However, the BART model does not capture the hierarchical structures of documents when generating a summary.

Neural machine translation has been improved by the capture of multiple granularities of information in input texts such as "phrases and words" and "words and characters". In particular, Transformer-based machine translation model has been improved by incorporating multi-granularity self-attention (MG-SA) (Hao et al., 2019), which considers the relationships between words and phrases by decomposing an input text into its elements using multiple granularity (i.e., words and phrases) and assigning each granular element (i.e., a word or a phrase) to a head in multi-head Self Attention Networks (SANs). This method enables interactions not only between words but also between phrases and words, through self-attentions.

Inspired by previous work, this paper proposes a new abstractive document summarization model, hierarchical BART (Hie-BART), which captures a document's hierarchical structures (i.e., sentence word structures) through the SANs of the BART model. Here, a document is divided into elements with word-level and sentence-level granularity, where each element is assigned to a head of the SANs layers of the BART encoder. Then, information with multi-granularity is captured by combining the output of the SANs layers, where the ratio of combining word-level and sentence level information hyperparameter.

Text summarization is the problem of creating a short, accurate, and fluent summary of a longer text document. Automatic text summarization methods are greatly needed to address the ever-growing amount of text data available online to both better help discover relevant information and to consume relevant information faster. This method enables interactions not only between words but also between phrases and words, through self-attentions.

1.2 Existing System

In existing system does not focus on Advanced NLP Techniques. End user does not get reliable summaries. In Existing system uses Extractive Text Summarization Technique Which does not give reliable summary. In this paper, we focus on improving the existing system of the user.

Disadvantages:

1. Summary is less accurate.
2. Time constraint is less.
3. Computational speed is more.
4. It is uses extractive summarization.

1.3 Proposed System

In the proposed system mainly focuses on providing a reliable summary. In the proposed system we are using Abstractive Text Summarization Which will provide the reliable summary. BART model is used.

Advantages:

- Time Constraint is less.
- Computational Speed is more.
- Accurate Summary.
- It uses Abstractive Summarization**.**

1.4 System Requirements

1.4.1 Hardware Requirements:

System type	:	intel corei7-7500UCPU 2.70gh
Cache memory	:	4 MB
RAM	:	12 GB
Hard Disc	:	8 GB

1.4.2 Software Requirements:

Operating system	:	windows 11, 64bit OS
Coding language	:	Python
Python distribution	:	Anaconda, Flask

2. LITERATURE SURVEY

Recent advancements in technology have enabled us to use ML models and perform exceptionally well in tasks unusually performed by humans. Higher computational power has empowered the use of complex sometimes non-linear models to be tasked on tasks such as text summarization, where they yield exceptionally high accuracies sometimes even better than humans.

Previous researches have introduced various summarization techniques which are cited in this section. Most of the researches concentrate on sentence extraction rather than generation for text summarization. Automatic text summarization as aptly mentioned in Mehdi Alla hyari et al. [4] is the task of automatically producing a concise and fluent summary without human intervention.

This is achieved by a machine using various techniques some of which incorporate NLP and Deep learning (neural networks) as the base. There are two base divisions, Extractive summarization and Abstractive summarization.

Qaiser, Shahzad & Ali, et al. [3], the use of Term Frequency Inverse Document Frequency (TF-IDF) is discussed in examining the relevance of key-words to documents in corpus. Extractive text summarization uses a statistical based approach to select important sentences or words from the document. Statistical approach can summarize the document using features like term frequency, location, and title, assigning weights to the keywords and then calculating the score of the sentence and selecting the highest scored sentence into the summary.

The study is focused on how the algorithm can be applied to a number of documents. First, the working principle and flow which should be followed for implementation of TF-IDF is illustrated. Secondly, to verify the findings from executing the algorithm, results are presented, and then strengths and weaknesses of TF-IDF algorithm are evaluated. This paper also tackles such weaknesses.

John X. Qiu, Hong-Jun Yoon, et al. [5], we see an implementation of Extractive summarization using convolutional Neural Network (CNN) for extracting ICDO-3 topographic codes from a corpus of breast and lung cancer pathology reports. The purpose was the extraction of information from the export and therefore helps in faster understanding of a report. However, the basic task was to extract words from the document and place them and not to consider the grammatical nuance, flow of words (sequences) and carry over the context.

Alexander M. Rush, Sumit Chopra, et al. [11], we see the use of Long Short Term Memory (LSTM) [11] as a base unit in an encoder decoder model for summarization of news articles. Ramesh Nallapati, Bowen Zhou, et al. [7] introduces a model called Sequence to Sequence (Seq2Seq) proposed in [10] which makes use of an encoder decoder arrangement with an attention mechanism for summary generation which again makes use of LSTM [11]

Abigail See, Peter J. Liu, et al [6], we see an implementation of a pointer generator network which utilizes a pointer network [9] with our traditional approach in [6] for a decoder. Alexander M. Rush, Sumit Chopra, et al. [11] proposes a more data driven approach as compared with the traditional generative approach. Using the model described in [7] as a base this paper draws a parallel between a Convolutional encoder a neural network language model (NNLM) encoder (as compared to a traditional LSTM [11]) and an attention based encoder.

2.1 Machine Learning

Machine learning is an application of artificial intelligence (AI) that provides systems the ability to automatically learn and improve from experience without being explicitly programmed. Machine learning focuses on the development of computer programs that can access data and use it learn for themselves.

There are important applications for text summarization in various NLP related tasks such as text classification, question answering, legal texts summarization, news summarization, and headline generation. Moreover, the generation of summaries can be integrated into these systems as an intermediate stage which helps to reduce the length of the document.

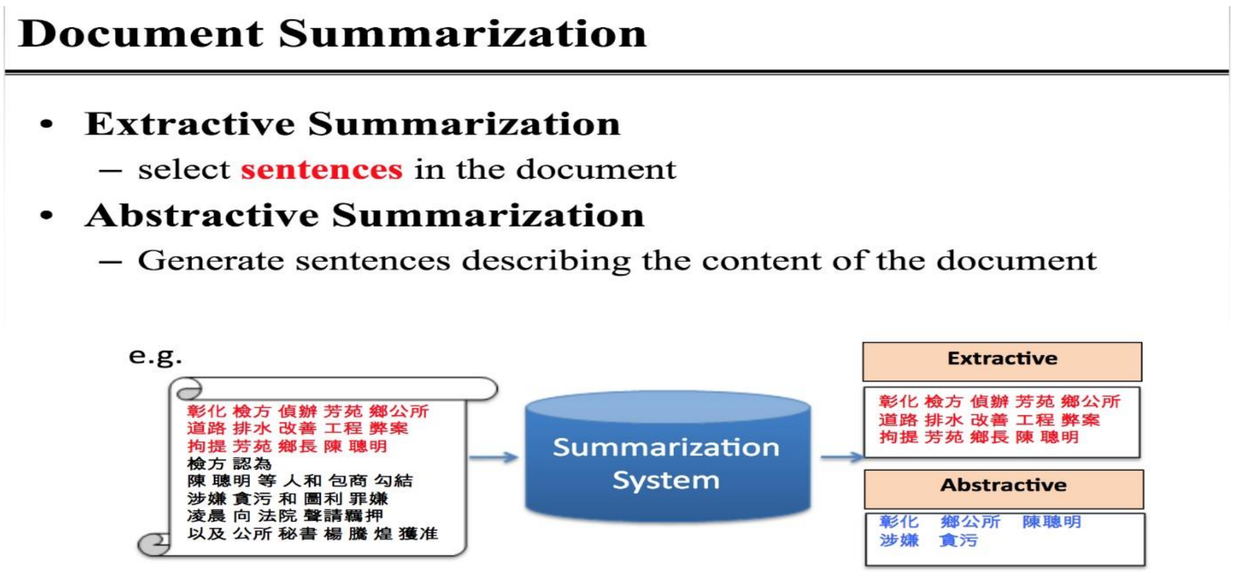
In the big data era, there has been an explosion in the amount of text data from a variety of sources. This volume of text is an inestimable source of information and knowledge which needs to be effectively summarized to be useful. This increasing availability of documents has demanded exhaustive research in the NLP area for automatic text summarization. Automatic text summarization is the task of producing a concise and fluent summary without any human help while preserving the meaning of the original text document.

It is very challenging, because when we as humans summarize a piece of text, we usually read it entirely to develop our understanding, and then write a summary highlighting its main points. Since computers lack human knowledge and language capability, it makes automatic text summarization a very difficult and non-trivial task.

Various models based on machine learning have been proposed for this task. Most of these approaches model this problem as a classification problem which outputs whether to include a sentence in the summary or not. Other approaches have used topic information, Latent Semantic Analysis (LSA), Sequence to Sequence models, Reinforcement Learning and Adversarial processes.

In general, there are two different approaches for automatic summarization:

- Extractive Summarization
- Abstractive Summarization



Document Summarization

- **Extractive Summarization**
  - select **sentences** in the document
- **Abstractive Summarization**
  - Generate sentences describing the content of the document

Figure 2.1: Types of Summarizations

Extractive Summarization:

Extractive summarization methods work just like that. It takes the text, ranks all the sentences according to the understanding and relevance of the text, and presents you with the most important sentences.

This method does not create new words or phrases, it just takes the already existing words and phrases and presents only that. You can imagine this as taking a page of text and marking the most important sentences using a highlighter.

Abstractive Summarization:

Abstractive summarization, on the other hand, tries to guess the meaning of the whole text and presents the meaning to you.

It creates words and phrases, puts them together in a meaningful way, and along with that, adds the most important facts found in the text. This way, abstractive summarization techniques are more complex than extractive summarization techniques and are also computationally more expensive.

Importance of Summarizing

Summarizing is of great importance for students to prosper in their careers as it improves their vocabulary and grammatical skills. Students who can adequately summarize a long text are good at focusing and extracting the main ideas. This is why summarizing is important for students.

Summarizing helps students to learn the technique of taking out the most important ideas from a text. They also learn to ignore irrelevant information that is present in the text, and students with these skills are capable of integrating the central ideas in a meaningful way from any theory or conceptual write-up. Students who are learning how to summarize, improve their memory abilities, and become more skilful in the process. Summarizing strategies is adopted in almost every area of studies or industry.

It acts as a great help for students to learn how to determine essential ideas and find out different details that can support those ideas and make them more useful. It helps the students to improve their focusing skills so that they can focus on phrases and keywords from the assigned long text.

Applications Of Summarization:

- Media Monitoring
- Newsletters
- Internal Document Workflow
- Legal Contrast Analysis
- Video Scripting
- Meetings and video Conferencing
- Automated Content Creation

2.2 Some machine learning methods

Machine learning algorithms are often categorized as supervised and unsupervised.

Supervised machine learning algorithms can apply what has been learned in thepast to new data using labeled examples to predict future events. Starting from the analysisof a known training dataset, the learning algorithm produces an inferred function to make predictions about the output values. The system is able to provide targets for any newinput after sufficient training. The learning algorithm can also compare its output with the correct, intended output and find errors in order to modify the model accordingly.

Unsupervised machine learning algorithms are used when the information used to train is neither classified nor labeled. Unsupervised learning studies how systems can infer a function to describe a hidden structure from unlabeled data. The system doesn't figure out the right output, but it explores the data and can draw inferences from datasetsto describe hidden structures from unlabeled data.

Reinforcement machine learning algorithms is a learning method that interacts with its environment by producing actions and discovers errors or rewards. Trial and error

search and delayed reward are the most relevant characteristics of reinforcement learning. This method allows machines and software agents to automatically determine the ideal behaviour within a specific context in order to maximize its performance. Simple reward feedback is required for the agent to learn which action is best. This is known as the reinforcement signal.

2.3 Applications of machine learning

1. Virtual Personal Assistants
2. Predictions while Commuting
3. Videos Surveillance
4. Social Media Services
5. Email Spam and Malware Filtering
6. Online Customer Support
7. Search Engine Result Refining
8. Product Recommendations
9. Online Fraud Detection.

3. DESIGN

3.1 Class Diagram

The class diagram depicts a static view of an application. It represents the types of objects residing in the system and the relationships between them. A class consists of its objects, and also it may inherit from other classes. A class diagram is used to visualize, describe, document various different aspects of the system, and also construct executable software code.

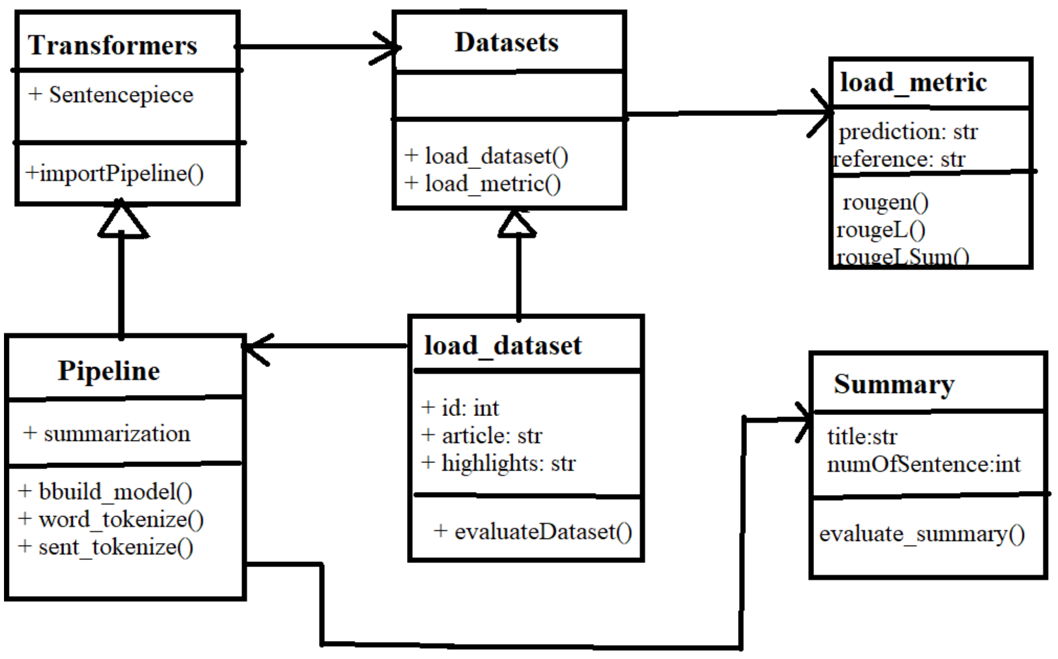


Figure: 3.1.1: Class Diagram

3.2 Activity Diagram

Activity Diagrams describe how activities are coordinated to provide a service which can be at different levels of abstraction. Typically, an event needs to be achieved by some operations, particularly where the operation is intended to achieve a number of different things that require coordination, or how the events in a single use case relate to one another, in particular, use cases where activities may overlap and require coordination. It is also suitable for modeling how a collection of use cases co-ordinate to represent business workflows.

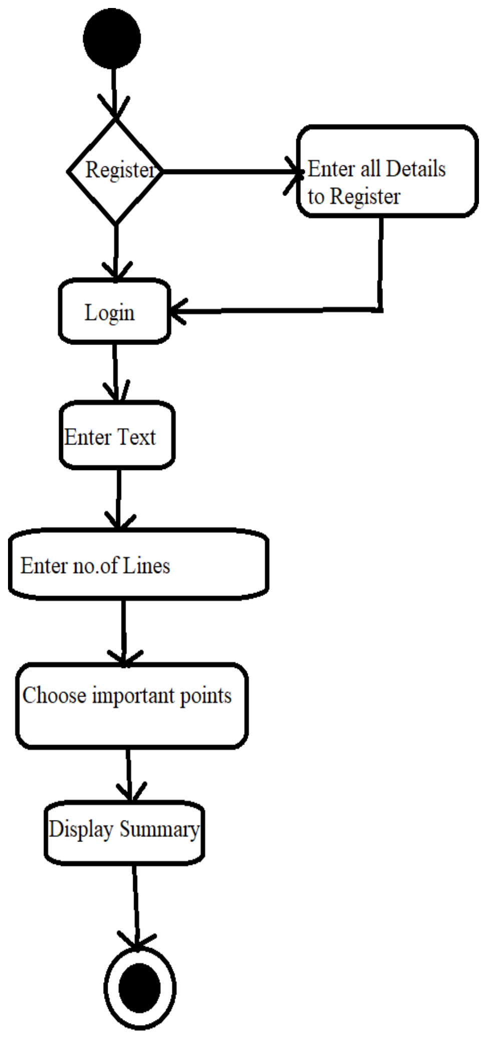


Figure: 3.2.1: Activity Diagram

3.3 Use Case Diagram

Use-case diagrams illustrate and define the context and requirements of either an entire system or the important parts of the system. You can model a complex system with a single use-case diagram, or create many use-case diagrams to model the components of the system. You would typically develop use-case diagrams in the early phases of a project and refer to them throughout the development process.

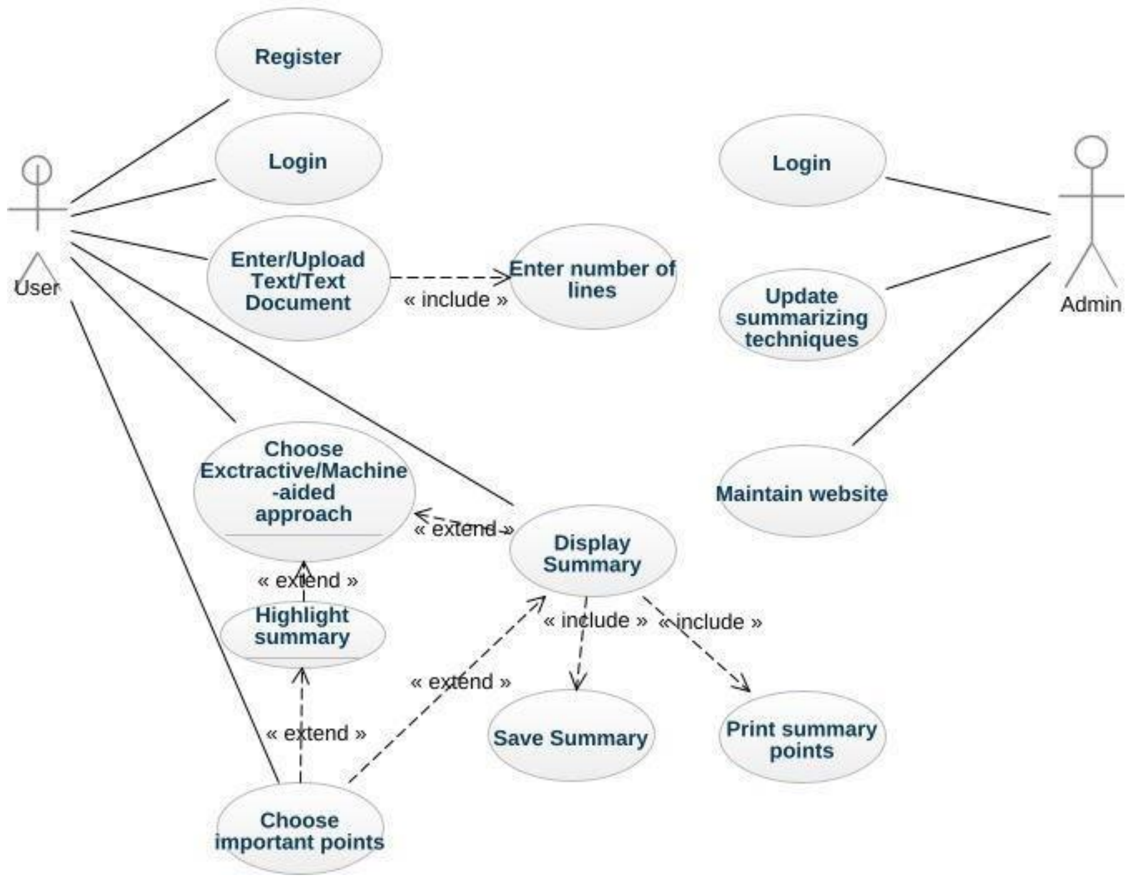


Figure: 3.3.1 Use Case Diagram

4. SYSTEM ANALYSIS

4.1 Hie-BART Architecture

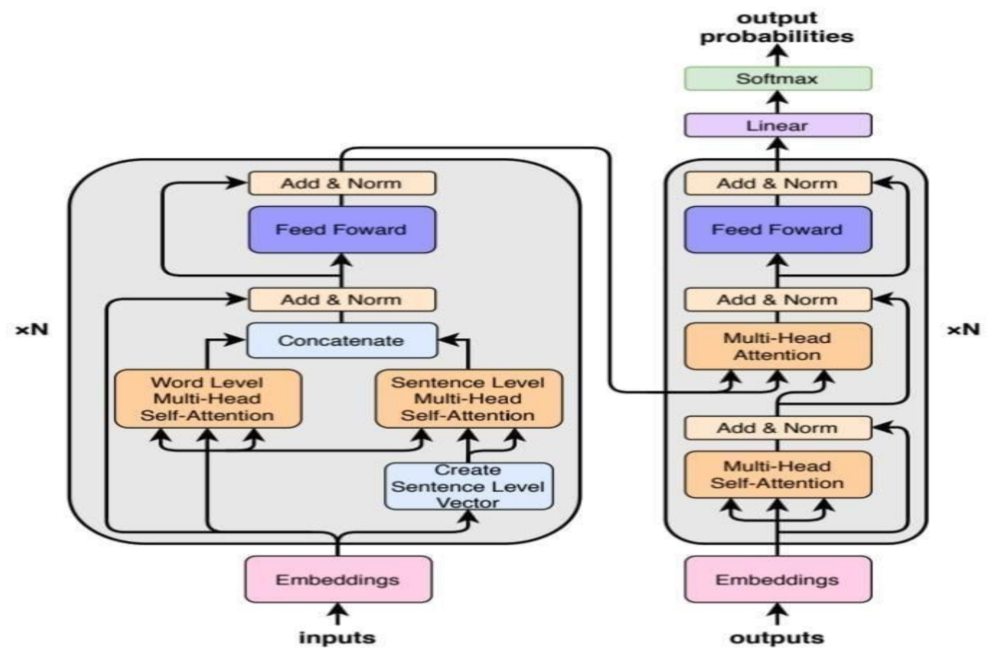


Figure 4.1.1: Architecture of Hie-BART

This is based on the Transformer model. The SANs in the encoder are divided into word and sentence levels and computed.

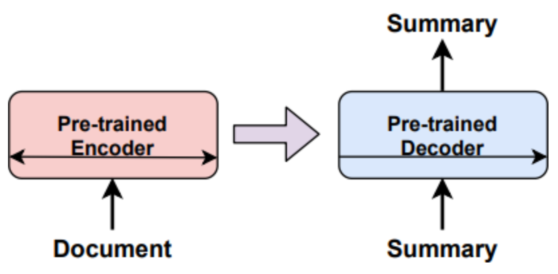


Figure 4.1.2 : Architecture of BART

4.2 BART Model:

BART is a sequence-to-sequence model trained as a denoising autoencoder. This means that a fine-tuned BART model can take a text sequence (for example, English) as input and produce a different text sequence at the output (for example, French). This type of model is relevant for machine translation (translating text from one language to another), question-answering (producing answers for a given question on a specific corpus), text summarization (giving a summary of or paraphrasing a long text document), or sequence classification (categorizing input text sentences or tokens). Another task is sentence entailment which, given two or more sentences, evaluates whether the sentences are logical extensions or are logically related to a given statement.

4.2.1 Five Pre-Training Techniques:

Five pre-training techniques are introduced: Token Masking, Sentence Permutation, Document Rotation, Token Deletion and Text Infilling.

Token Masking:

Random tokens in a sentence are replaced with [MASK]. The model learns how to predict the single token based on the rest of the sequence.

Sentence Permutation:

Sentences (separated by full stops) are permuted randomly. This helps the model to learn the logical entailment of sentences.

Document Rotation:

The document is rearranged to start with a random token. The content before the token is appended at the end of the document. This gives insights into how the document is typically arranged and how the beginning or ending of a document looks like.

Token Deletion:

Random tokens are deleted. The model must learn to predict the token content and find the position where the token was deleted from.

Text Infilling:

A fixed number of contiguous tokens are deleted and replaced with a single [MASK] token. The model must learn the content of the missing tokens and the number of tokens.

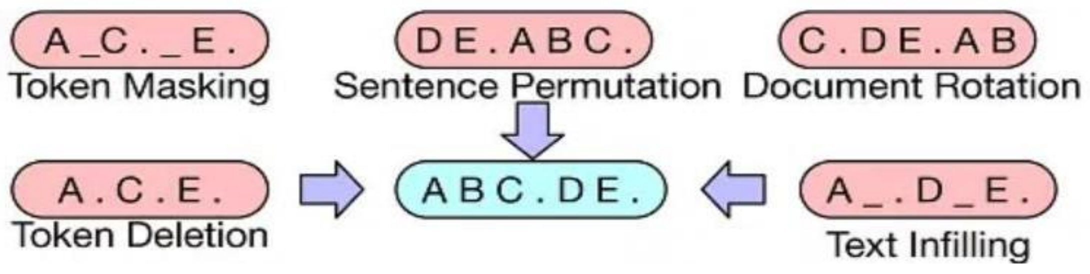


Figure 4.2.1.1: Pre-Training Techniques

4.2.2 Multi-Granularity Self-Attention (MG-SA)

MG-SA (Hao et al., 2019) is used to capture multigranularity information from an input text by dividing the input into elements with several types of granularity and preparing heads of multi-head SANs for each type of granularity. Provided with the word-level matrix H, which is an input to the SANs, this method first generates a phrase-level matrix Hg representing phrase-level information, as follows:

$$Hg = Fh(H)$$

where Fh(▪) is a function that generates a phrase level matrix for the h-th head. Specifically, a phrase-level matrix is generated by running a max pooling operation on word-level vectors in a wordlevel matrix. After a phrase-level matrix is generated, SANs perform the following computations:

$$Q^{h}, K^{h}, V^{h} = HW^{h}_{Q},\ H_{g}W^{h}_{K},\ H_{g}W^{h}_{V} \ (1)$$

$$O^{h} = \mathbf{ATT}(Q^{h}, K^{h})V^{h}, \ (2)$$

where $Q^h \in R^{n\times dh}$, $K^h \in R^{p\times dh}$, $V h \in R^{p\times dh}$ are respectively the query, key, and value representations, $W^h_Q, W^h_K, W^h_V \in R^{d\times dh}$ are parameter matrices, and d, dh, n, and p are the dimensions of the hidden layer, one head, a word vector, and a phrase vector, respectively. In addition, ATT(X,Y) is a function that calculates the attention weights of X and Y. From these computations, the output Oh of each head in the SANs is generated. Then, the output of MG-SA is generated by concatenating the outputs from all heads: MG-SA(H) = [O1 , ..., ON]. The outputs of each head Oh contain information between words or between words and phrases. Thus, in addition to relationships between words, the relationships between words and phrases can be captured with MG-SA.

4.2.3 Sentence Level Vector Layer:

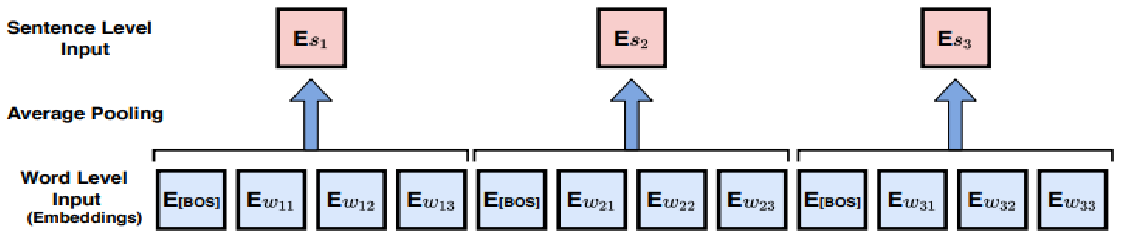


Figure 4.2.3.1: Behavior of The Sentence Level Vector Layer.

The behavior of the create sentence level vector layer is shown in Figure 3. Ewij and E[BOS] are embedded vectors for word wij (j-th word in the i-th sentence) and [BOS] token, respectively. Esi is the sentence-level embedded vector for the i-th sentence si . The create sentence level vector layer uses average pooling to generate a sentence-level vector from word-level vectors. Given the word sequence W = (w1, ..., wN), it is divided into sentences

S = (s1, ..., sM)

where N is the total number of words, M is the total number of sentences, and each si is the i-th sentence consisting of a word subsequence wi1, . . . wiNi , where Ni is the total number of words in the sentence. For each element of S, we apply average pooling as follows: gm = AVG(sm), where the AVG(▪) is average pooling. From this formula, G = (g1, …, gM) is generated. Each element of W, S, and G is an embedded vector. G is forwarded to the sentence level SANs as its input.

4.2.4 Concatenate Layer:

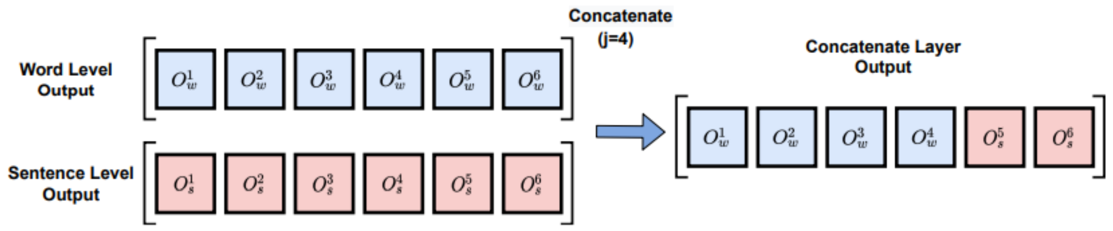


Figure 4.2.4.1: Behavior of the Concatenate Layer

The outputs of each of the word-level and sentence-level SANs are combined in the concatenate layer. The outputs of the word-level and sentence-level SANs layer are as follows:

$$\mathbf{SANs(W) = [O_1^w, \ldots, O_H^w] = O_{ALL}^w}$$

$$\mathbf{SANs(G) = [O_1^s, \ldots, O_H^s] = O_{ALL}^s}$$

where H is the number of heads, $[O_1^w, \ldots, O_H^w] = O_{ALL}^w$ is the output of the word-level SANs, consisting of the word-level head's outputs, and $[O_1^s, \ldots, O_H^s] = O_{ALL}^s$ is the output of the sentence-level SANs, consisting of the sentence level head's outputs. The outputs of these word/sentence-level SANs are combined as follows:

$$\mathbf{CONCAT(O_{ALL}^w, O_{ALL}^s, j) = [O_1^w, \ldots, O_j^w, O_{j+1}^s, \ldots, O_H^s]}$$

where CONCAT(X, Y, j) is a function that concatenates X and Y at the join point j of the multi heads. In the combined multi-head, the heads from 1 to j are word-level outputs, and the heads from j + 1 to H are sentence-level outputs. Figure 4 shows an example of the behavior of the concatenate layer in Hie-BART, where the number of heads of the multi-head is 6 and the join point j = 4. The output of the wordlevel SANs $[O_1^w, \ldots, O_6^w]$ and the output of the sentence-level SANs $[O_1^s, \ldots, O_6^s]$ are joined at the join point j = 4, resulting in the output $[O_1^w, O_2^w, O_3^w, O_4^w, O_5^s, O_6^s]$. The output of the concatenate layer is forwarded to the feed-forward layer in the encoder.

4.3 Importance of machine learning in Text Summarization

Text summarization refers to the technique of shortening long pieces of text. The intention is to create a coherent and fluent summary having only the main points outlined in the document. Automatic text summarization is a common problem in machine learning and natural language processing (NLP). Machine learning models are usually trained to understand documents and distill the useful information before outputting the required summarized texts.

4.4 Implementation of machine learning using Python

Python is a popular programming language. It was created in 1991 by Guido van Rossum. It is used for:

1.Web development (server-side)

2.Software development

3.Mathematics

4.System scripting.

The most recent major version of Python is Python 3. However, Python 2, although not being updated with anything other than security updates, is still quite popular.

It is possible to write Python in an Integrated Development Environment, such as Thonny, Pycharm, Netbeans or Eclipse, Anaconda which are particularly useful when managing larger collections of Python files. Python was designed for its readability. Python uses new lines to complete a command, asopposed to other programming languages which often use semicolons or parentheses. Python relies on indentation, using whitespace, to define scope; such as the scope of loops,functions and classes. Other programming languages often use curly-brackets for this purpose.

In the older days, people used to perform Machine Learning tasks manually by coding all the algorithms and mathematical and statistical formula. This made the process time consuming, tedious and inefficient. But in the modern days, it is become very much easy and efficient compared to the olden days by various python libraries, frameworks, and modules. Today, Python is one of the most popular programming languages for this task and it has replaced many languages in the industry, one of the reason is its vast collection of libraries. Python libraries that used in Machine Learning are:

1.Numpy

2.Scipy

3.Scikit-learn

4.Pandas

5.Matplotlib

NumPy is a very popular python library for large multi-dimensional array and matrix processing, with the help of a large collection of high-level mathematical functions. It is very useful for fundamental scientific computations in Machine Learning. It is particularly useful for linear algebra, Fourier transform, and random number capabilities. High-end libraries like TensorFlow uses NumPy internally for manipulation of Tensors.

SciPy is a very popular library among Machine Learning enthusiasts as it contains different modules for optimization, linear algebra, integration and statistics. There is a difference between the SciPy library and the SciPy stack. The SciPy is one of the core packages that make up the SciPy stack. SciPy is also very useful for image manipulation.

Skikit-learn is one of the most popular Machine Learning libraries for classical Machine

Learning algorithms. It is built on top of two basic Python libraries, NumPy and SciPy. Scikit-learn supports most of the supervised and unsupervised learning algorithms. Scikit learn can also be used for data-mining and data-analysis, which makes it a great tool who is starting out with Machine Learning.

Pandas is a popular Python library for data analysis. It is not directly related to Machine Learning. As we know that the dataset must be prepared before training. In this case, Pandas comes handy as it was developed specifically for data extraction and preparation. It provides high-level data structures and wide variety tools for data analysis. It provides many inbuilt methods for groping, combining and filtering data.

Matpoltlib is a very popular Python library for data visualization. Like Pandas, it is not directly related to Machine Learning. It particularly comes in handy when a programmer wants to visualize the patterns in the data. It is a 2D plotting library used for creating 2D graphs and plots. A module named pyplot makes it easy for programmers for plotting as it provides features to control line styles, font properties, formatting axes, etc. It provides various kinds of graphs and plots for data visualization, histogram, error charts, bar chats, etc.

4.5 Scope of the project

The scope of this system is to generate the summary of a large article or document present in datasets, train the model using thelarge quantity of data present in datasets and predict whether the summary is generating or not on new data during testing.

4.5.1 Analysis

The dataset contains 3 attributes which are used to generate the summary of large article using bart model:

1. Article_id
2. Article
3. Highlights

Article_id: Holds the numerical data which carries the serial unique id's for all the articles present in the dataset.

Article: Article attribute contains the large document or article which we need to summarize**.**

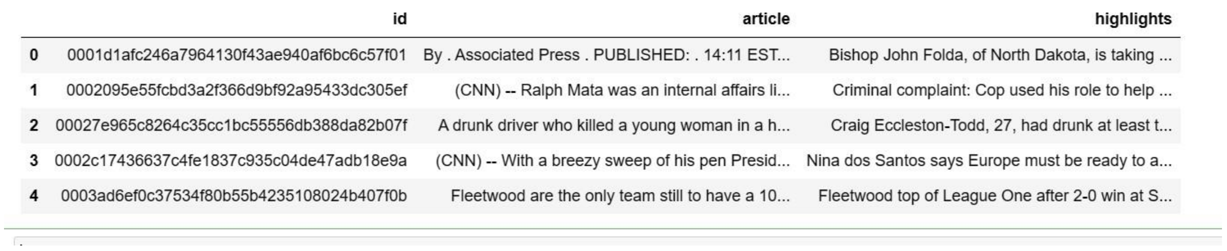
Highlights: The Highlights attribute contains the summary which is generated by the ml model which is further used as a reference for checking the accuracy of our summary.

4.5.2 Dataset:

We used the CNN/Daily Mail dataset1 (Hermann et al., 2015), a summary corpus of English news articles, consisting of 287,226 training pairs, 13,368 validation pairs, and 11,490 test pairs. On average, the source documents and summary sentences have 781 and 56 tokens, respectively. For data preprocessing, we followed the instruction provided in the CNN/Daily Mail dataset1 and fairseq2.

Dataset Link:

https://www.kaggle.com/datasets/gowrishankarp/newspaper-text-summarization-cnn-dailymail



	id	article	highlights
0	0001d1afc246a7964130f43ae940af6bc6c57f01	By . Associated Press . PUBLISHED: . 14:11 EST...	Bishop John Folda, of North Dakota, is taking ...
1	0002095e55fcbd3a2f366d9bf92a95433dc305ef	(CNN) -- Ralph Mata was an internal affairs li...	Criminal complaint: Cop used his role to help ...
2	00027e965c8264c35cc1bc55556db388da82b07f	A drunk driver who killed a young woman in a h...	Craig Eccleston-Todd, 27, had drunk at least t...
3	0002c17436637c4fe1837c935c04de47adb18e9a	(CNN) -- With a breezy sweep of his pen Presid...	Nina dos Santos says Europe must be ready to a...
4	0003ad6ef0c37534f80b55b4235108024b407f0b	Fleetwood are the only team still to have a 10...	Fleetwood top of League One after 2-0 win at S...

Fig 3.5.2.1 Dataset

4.5.3 Parameters:

We used the pre-trained BART model "bart-arge", provided in fairseq2 for Hie-BART. The hyper parameters for BART and Hie-BART were determined for the validation set; the gradient accumulation parameter (update-freq) was 10, the total number of training steps was 20,000, and the number of multi-heads was set to 16. The ratio of the number of combined heads of output in word-level and sentence-level SANs was set to "Word: Sentence = 14:2" for Hie-BART. We followed fairseq's settings2 for the other hyper parameters. In our environments, the model had 406,291,456 parameters for Hie-BART and 406,290,432 parameters for BART.

5. Machine Learning Algorithms For Summarization

As there are two types of approaches for text summarization such as Extractive Text Summarization and Abstractive Text Summarization. To generate summary with these two approaches we several popular machine learning algorithms.

5.1 Approaches for Extractive Text Summarization:

We can perform the Extractive Text Summarization using these Machine Learning Models such as:

1. Using Gensim with TextRank
2. Text Summarization with Sumy
3. Using LexRank
4. Using LSA (Latent semantic analysis)
5. Using KL-Sum

5.1.1 Gensim With TextRank Algorithm:

Gensim is a very handy python library for performing NLP tasks. The text summarization process using gensim library is based on **TextRank Algorithm**

TextRank Algorithm: TextRank is an extractive summarization technique. It is based on the concept that words which occur more frequently are significant. Hence , the sentences containing highly frequent words are important.

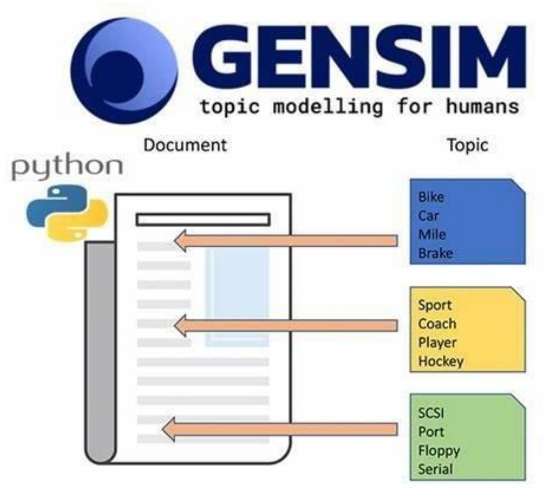


Figure 5.1.1.1 : Gesim Alogorithm

5.1.2 LexRank Algorithm:

A sentence which is similar to many other sentences of the text has a high probability of being important. The approach of LexRank is that a particular sentence is recommended by other similar sentences and hence is ranked higher. Higher the rank, higher is the priority of being included in the summarized text.

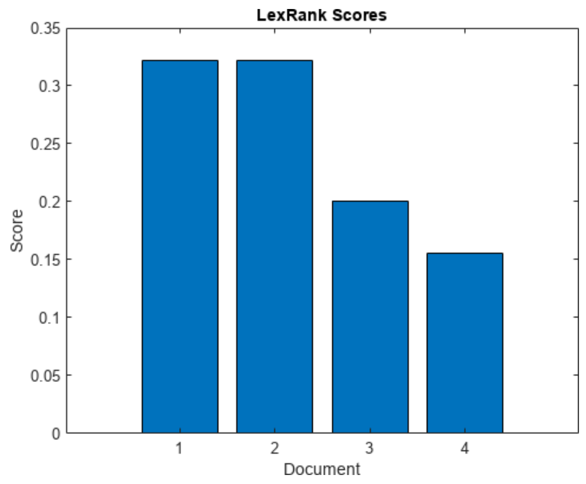


Figure 5.1.2.1 : LexRank Algorithm Graph

5.1.3 LSA (Latent semantic analysis) Algorithm:

Latent Semantic Analysis is a unsupervised learning algorithm that can be used for extractive text summarization. It extracts semantically significant sentences by applying singular value decomposition (SVD) to the matrix of term-document frequency.

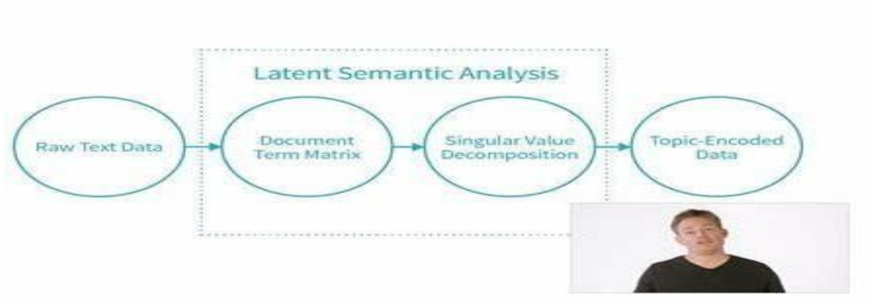


Figure 5.1.3.1 : Latent Semantic Analysis Algorithm

5.1.4 KL-Sum Algorithm:

It selects sentences based on similarity of word distribution as the original text. It aims to lower the KL-divergence criteria. It uses greedy optimization approach and keeps adding sentences till the KL-divergence decreases.

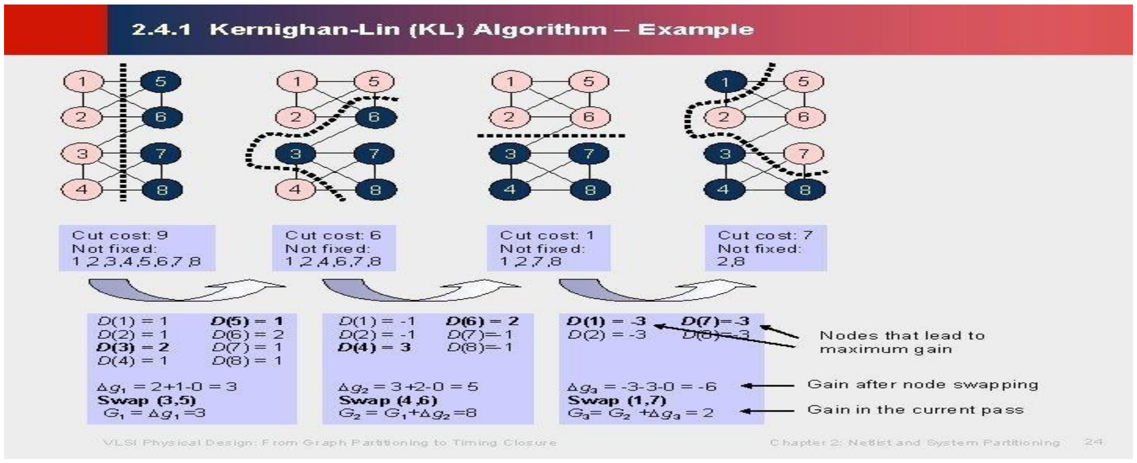

2.4.1 Kernighan-Lin (KL) Algorithm – Example
Cut cost: 9 Not fixed: 1,2,3,4,5,6,7,8
Cut cost: 6 Not fixed: 1,2,4,6,7,8
Cut cost: 1 Not fixed: 1,2,7,8
Cut cost: 7 Not fixed: 2,8
D(1) = 1 D(2) = 1 D(3) = 2 D(4) = 1 D(5) = 1 D(6) = 2 D(7) = 1 D(8) = 1
Δg1 = 2+1-0 = 3 Swap (3,5) G1 = Δg1 =3
D(1) = -1 D(2) = -1 D(4) = 3 D(6) = 2 D(7)= 1 D(8)= 1
Δg2 = 3+2-0 = 5 Swap (4,6) G2 = G1+Δg2 =8
D(1) = -3 D(2) = -3 D(7)=-3 D(8)=-3
Δg3 = -3-3-0 = -6 Swap (1,7) G3= G2 +Δg3 = 2
Nodes that lead to maximum gain
Gain after node swapping
Gain in the current pass
VLSI Physical Design: From Graph Partitioning to Timing Closure
Chapter 2: Netlist and System Partitioning 24

Figure 5.1.4.1: KL-Sum Algorithm

5.2 Approaches for Abstractive Text Summarization:

We can perform the Abstractive Text Summarization using these Machine Learning Models such as:

1. Using T5 Transformers
2. Using BART Model
3. Using GPT-2 Model
4. Using Pegasus Model

5.2.1 T5 Transformer Model:

T5 is an encoder-decoder model. It converts all language problems into a text-to-text format. First, you need to import the tokenizer and corresponding model through below command. It is preferred to use T5ForConditionalGeneration model when the input and output are both sequences.

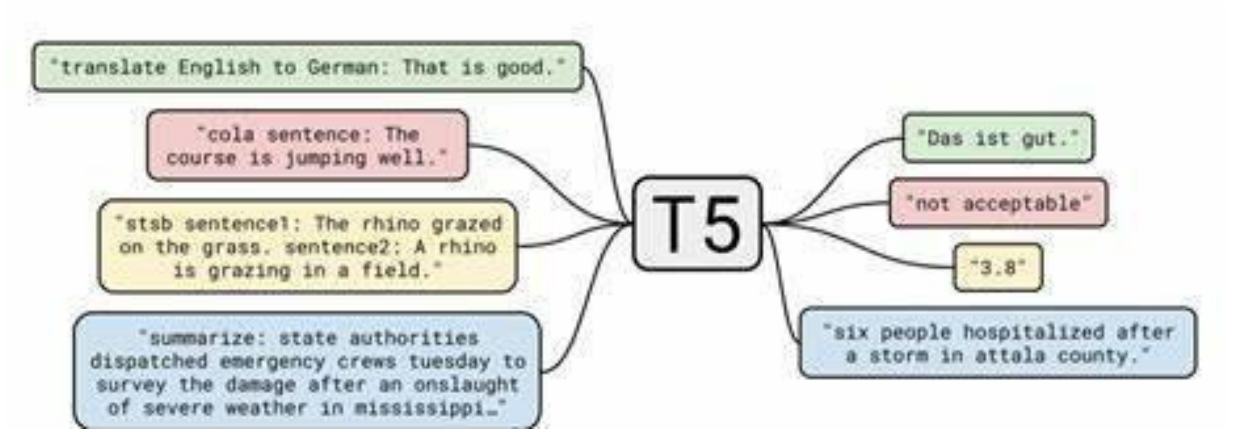

"translate English to German: That is good."
"cola sentence: The course is jumping well."
"stsb sentence1: The rhino grazed on the grass. sentence2: A rhino is grazing in a field."
"summarize: state authorities dispatched emergency crews tuesday to survey the damage after an onslaught of severe weather in mississippi…"
T5
"Das ist gut."
"not acceptable"
"3.8"
"six people hospitalized after a storm in attala county."

Figure: 5.2.1.1 :T5 Transformer Model

5.2.2 BART Model:

BART is a sequence-to-sequence model trained as a denoising autoencoder. This means that a fine-tuned BART model can take a text sequence (for example, English) as input and produce a different text sequence at the output.

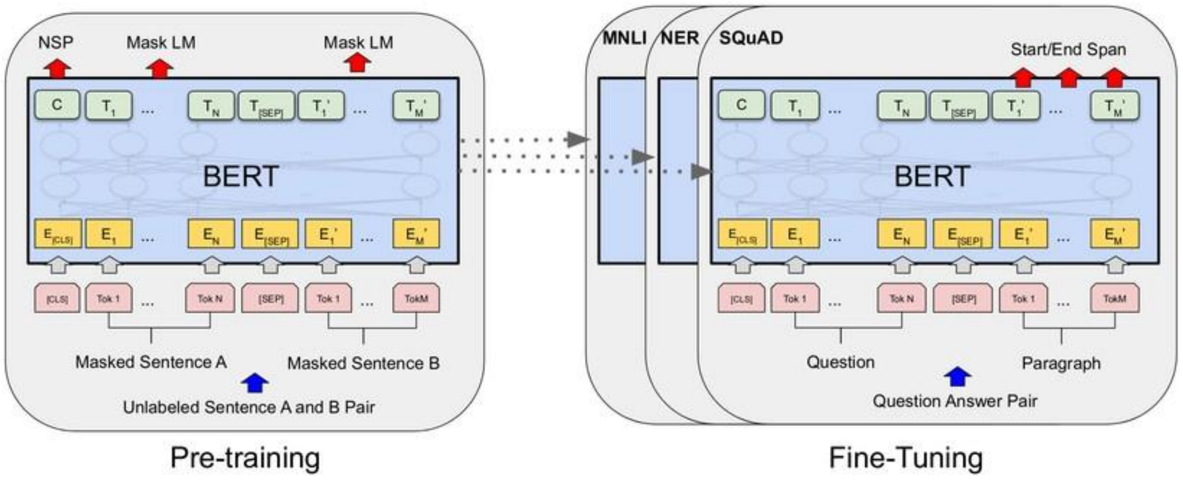
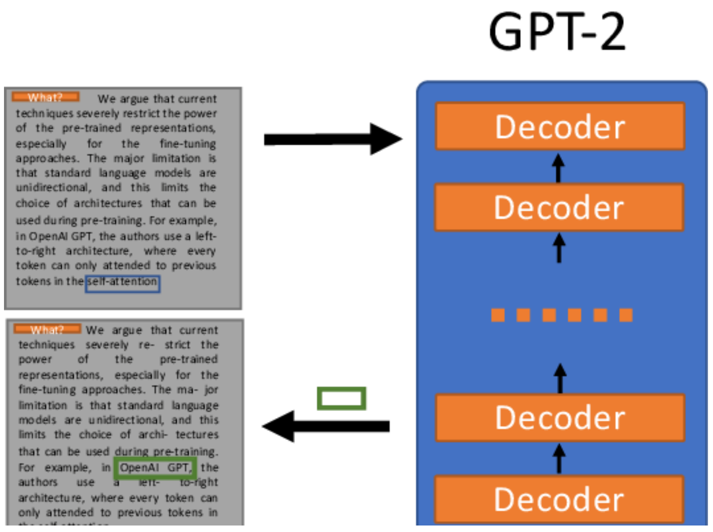


Figure: 5.2.2.1: BART Transformer Model

5.2.3 GPT-2 Model:

GPT-2 is a model with absolute position embeddings so it’s usually advised to pad the inputs on the right rather than the left.GPT-2 was trained with a causal language modeling (CLM) objective and is therefore powerful at predicting the next token in a sequence. Leveraging this feature allows GPT-2 to generate syntactically coherent text as it can be observed in the *run_generation.py* example script. The PyTorch models can take the *past* as input, which is the previously computed key/value attention pairs. Using this *past* value prevents the model from re-computing pre-computed values in the context of text generation.



We argue that current techniques severely restrict the power of the pre-trained representations, especially for the fine-tuning approaches. The major limitation is that standard language models are unidirectional, and this limits the choice of architectures that can be used during pre-training. For example, in OpenAI GPT, the authors use a left-to-right architecture, where every token can only attended to previous tokens in the self-attention

We argue that current techniques severely re- strict the power of the pre-trained representations, especially for the fine-tuning approaches. The ma- jor limitation is that standard language models are unidirectional, and this limits the choice of archi- tectures that can be used during pre-training. For example, in OpenAI GPT, the authors use a left- to-right architecture, where every token can only attended to previous tokens in

Figure: 5.2.3.1 GPT-2 Transformer Model

5.2.4 Pegasus Model:

in PEGASUS, **complete sentences are removed** from a document (i.e. they are ‘masked’), and the **model is trained to predict these sentences** as shown in the figure. The authors admit that this task seems nearly impossible, even for humans for a matter of fact. But such training elicits a higher sense of understanding for the generation of sentences that have an instance of the original document; thus supporting their assumption. This task is coined as **Gap Sentence Generation (GSG).**

A State-of-the-Art Model for Abstractive Text Summarization Students are often tasked with reading a document and producing a summary.

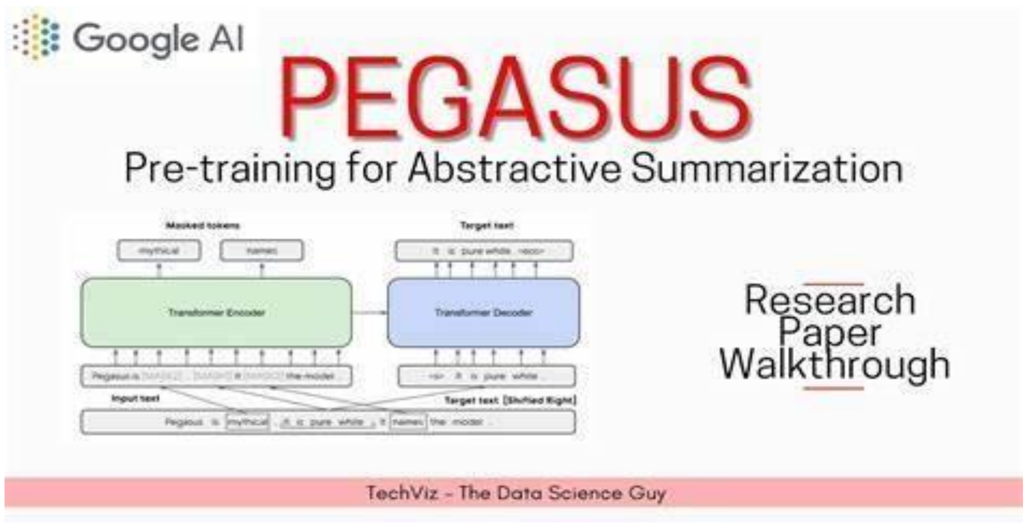

Google AI
PEGASUS
Pre-training for Abstractive Summarization
Masked tokens
Target text
Transformer Encoder
Transformer Decoder
Input text
Target text (Shifted Right)
Research Paper Walkthrough
TechViz - The Data Science Guy

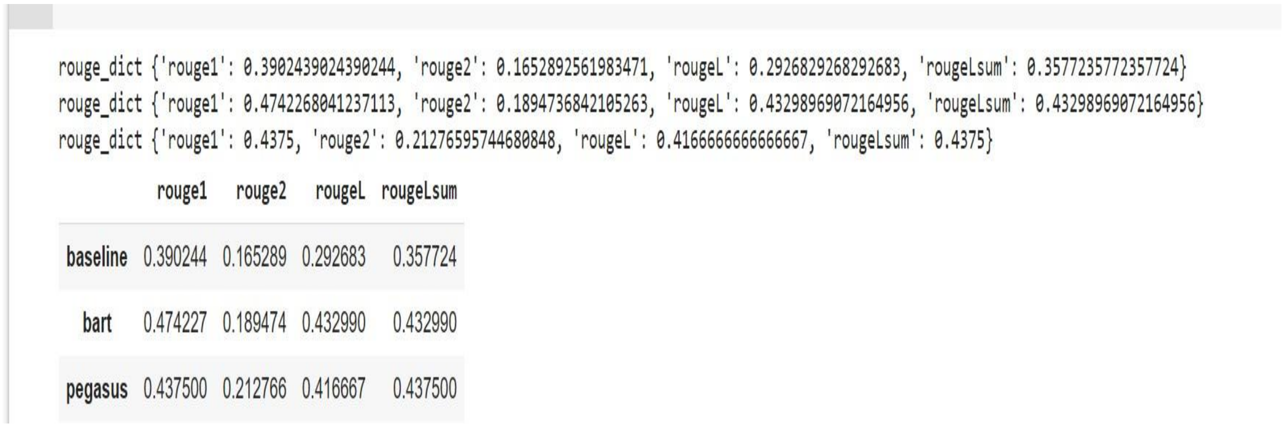
Figure 5.2.4.1: Pegasus Transformer Model

6. IMPLEMENTATION

6.1 BACKEND:

```
import numpy as npimport pandas as pdimport stringimport nltk
from nltk.tokenize import sent_tokenize, word_tokenize
! pip install transformers[sentencepiece] datasets rouge_scorefrom transformers import pipelie
from datasets import load_dataset,load_metricfrom transformers AutoTokenizer
from datasets import load_dataset! pip install pipelinepip install --upgrade pip
pip install tensorflowpip install numpy –upgradepip install --upgrade Pillow
dataset=load_dataset("cnn_dailymail",version="3.0.0")print(f"Features in cnn_dailymail : {dat
aset['train'].column_names}")sample_text=dataset['train'][1]['article'][:1000]summaries={}
sample_texthigh=dataset['train'][1]['highlights'][:1000]highdef summary(text):
return "\n".join(sent_tokenize(text)[:3])summaries['baseline']=summary(sample_text)
from transformers import pipelinepipe=pipeline("summarization",model="facebook/bart-large-
cnn")pipe_out=pipe(sample_text)pipe_outsummaries['bart']="\n".join(sent_tokenize(pipe_out[
0]["summary_text"]))summaries['bart']pipe=pipeline("summarization",model="google/pegasus
-cnn_dailymail")pipe_out=pipe(sample_text)
summaries['pegasus']="\n".join(sent_tokenize(pipe_out[0]["summary_text"])summaries['pegas
us']from datasets import load_metricrouge_names=["rouge1","rouge2","rougeL","rougeLsum"]
reference=dataset['test'][1]['highlights']records=[]for i in summaries:rouge_metric.add(predictio
n=summaries[i],reference=reference)score=rouge_metric.compute()rouge_dict=dict((rn,score[r
n].mid.fmeasure) for rn in rouge_names)print('rouge_dict',rouge_dict)records.append(rouge_di
ct)pd.DataFrame.from_records(records,index=summaries.keys() )reference=dataset['train'][1]['
highlights']for i in summaries:rouge_metric.add(prediction=summaries[i],reference=reference)
score=rouge_metric.compute()rouge_dict=dict((rn,score[rn].mid.fmeasure) for rn in rouge_na
mes)print('rouge_dict',rouge_dict)records.append(rouge_dict)
pd.DataFrame.from_records(records,index=summaries.keys() )
```

6.4 Result Analysis:



```
rouge_dict {'rouge1': 0.3902439024390244, 'rouge2': 0.1652892561983471, 'rougeL': 0.2926829268292683, 'rougeLsum': 0.3577235772357724}
rouge_dict {'rouge1': 0.4742268041237113, 'rouge2': 0.1894736842105263, 'rougeL': 0.43298969072164956, 'rougeLsum': 0.43298969072164956}
rouge_dict {'rouge1': 0.4375, 'rouge2': 0.21276595744680848, 'rougeL': 0.4166666666666667, 'rougeLsum': 0.4375}
```

	rouge1	rouge2	rougeL	rougeLsum
baseline	0.390244	0.165289	0.292683	0.357724
bart	0.474227	0.189474	0.432990	0.432990
pegasus	0.437500	0.212766	0.416667	0.437500

Figure: 6.4.1: Accuracy of Models

Comparison of Accuracy(rouge1, rouge2,rougeL,rougeLsum) on Models such as Baseline that means default model, BART model and Pegasus Model .

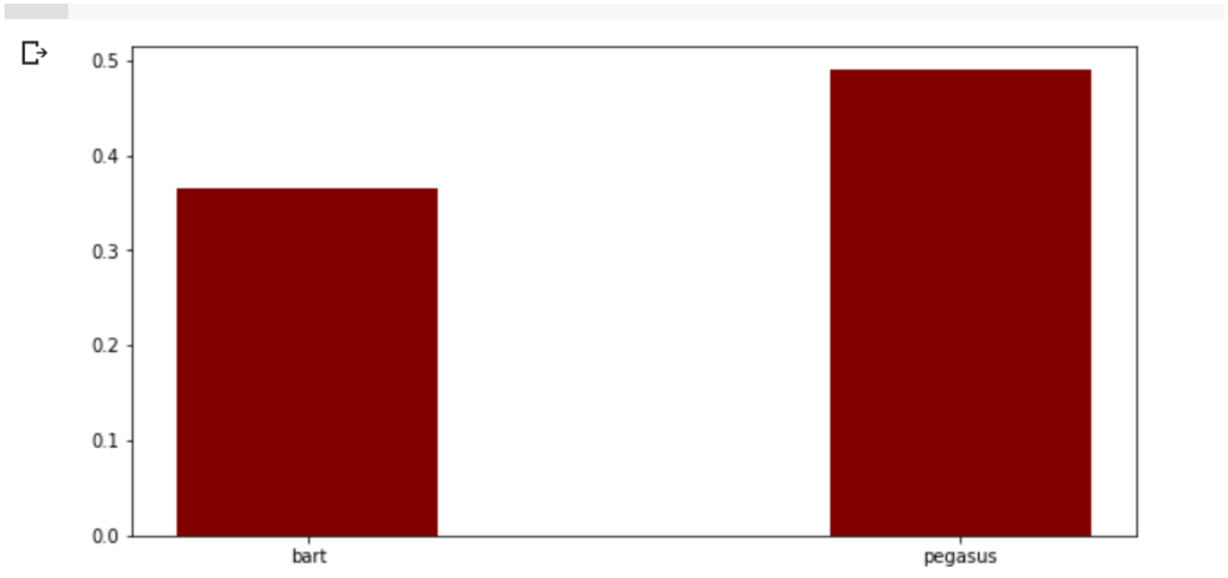


Figure 6.4.2: Comparision Through Graph

6.4.1 Rouge (Recall-Oriented Understudy for Gisting Evaluation):

It is a set of metrics and a software package specifically designed for evaluating automatic summarization, but that can be also used for machine translation. The metrics compare an automatically produced summary or translation against reference (high-quality and human-produced) summaries or translations.

Rouge-N:

It measures the number of matching n-grams between the model-generated text and a human-produced reference.

ROUGE-1:

ROUGE-1 precision can be computed as the ratio of the number of unigrams in *C* that appear also in *R* (that are the words "the", "cat", and "the"), over the number of unigrams

ROUGE-1 precision = 3/5 = 0.6

ROUGE-1 recall can be computed as the ratio of the number of unigrams in *R* that appear also in *C* (that are the words "the", "cat", and "the"), over the number of unigrams in *R*.

ROUGE-1 recall = 3/6 = 0.5

ROUGE-1 F1-score can be directly obtained from the ROUGE-1 precision and recall using the standard F1-score formula.

ROUGE-1 F1-score = 2 * (precision * recall) / (precision + recall) = 0.36

ROUGE-2:

ROUGE-2 precision is the ratio of the number of 2-grams in *C* that appear also in *R* (only the 2-gram "the cat"), over the number of 2-grams in *C*.

ROUGE-2 precision = 1/4 = 0.25

ROUGE-2 recall is the ratio of the number of 2-grams in *R* that appear also in *C* (only the 2-gram "the cat"), over the number of 2-grams in *R*.

ROUGE-2 recall = 1/5 = 0.20

ROUGE-2 F1-score is:

ROUGE-2 F1-score = 2 * (precision * recall) / (precision + recall) = 0.22

Rouge-L:

ROUGE-L is based on the longest common subsequence (LCS) between our model output and reference, i.e. the longest sequence of words (not necessarily consecutive, but still in order) that is shared between both. A longer shared sequence should indicate more similarity between the two sequences.

ROUGE-L precision is the ratio of the length of the LCS, over the number of unigrams in *C*.

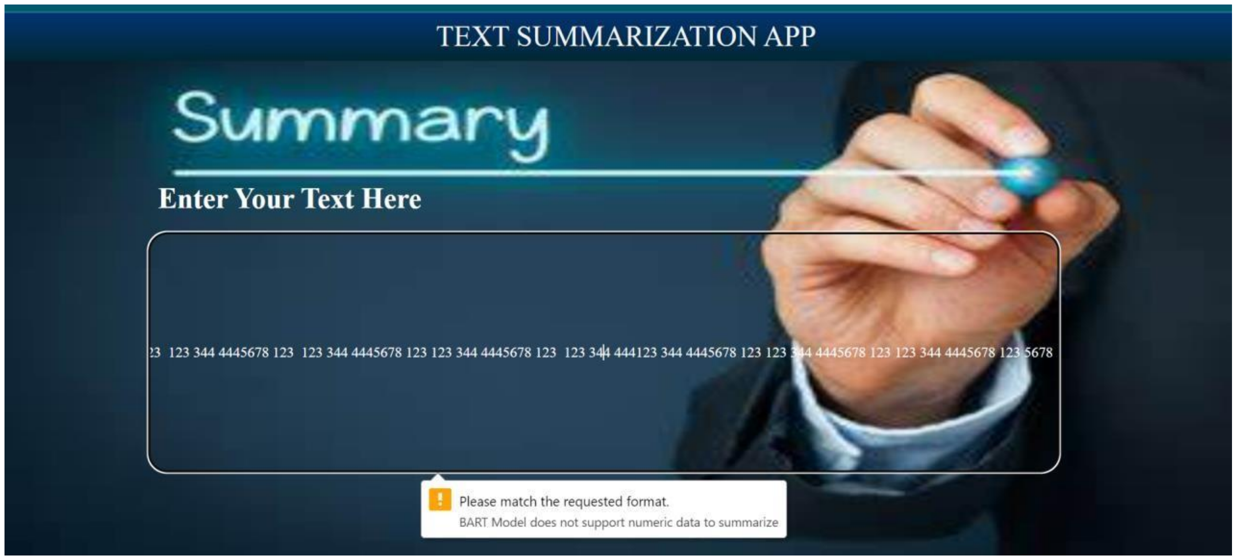
ROUGE-L precision = 3/5 = 0.6

ROUGE-L recall = 3/6 = 0.5

ROUGE-L F1-score = 2 * (precision * recall) / (precision + recall) = 0.55

7. SYSTEM TESTING

7.1 Test Case 1

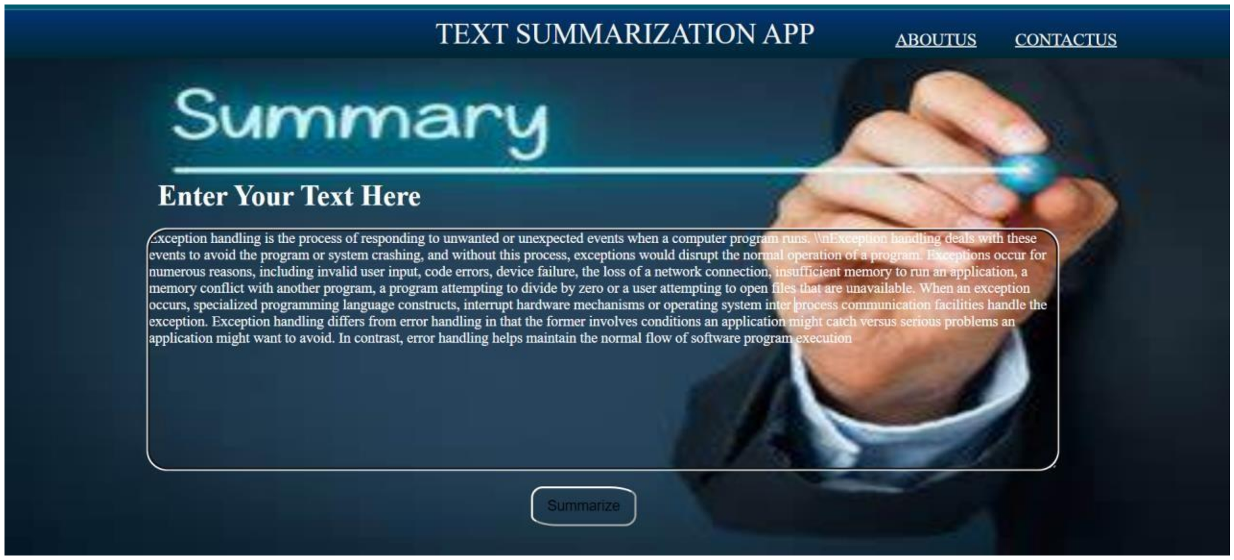


```
23 123 344 4445678 123 123 344 4445678 123 123 344 4445678 123 123 344 444123 344 4445678 123 123 344 4445678 123 123 344 4445678 123 5678
```

Please match the requested format.
BART Model does not support numeric data to summarize

Figure: 7.1.1 Invalid Input

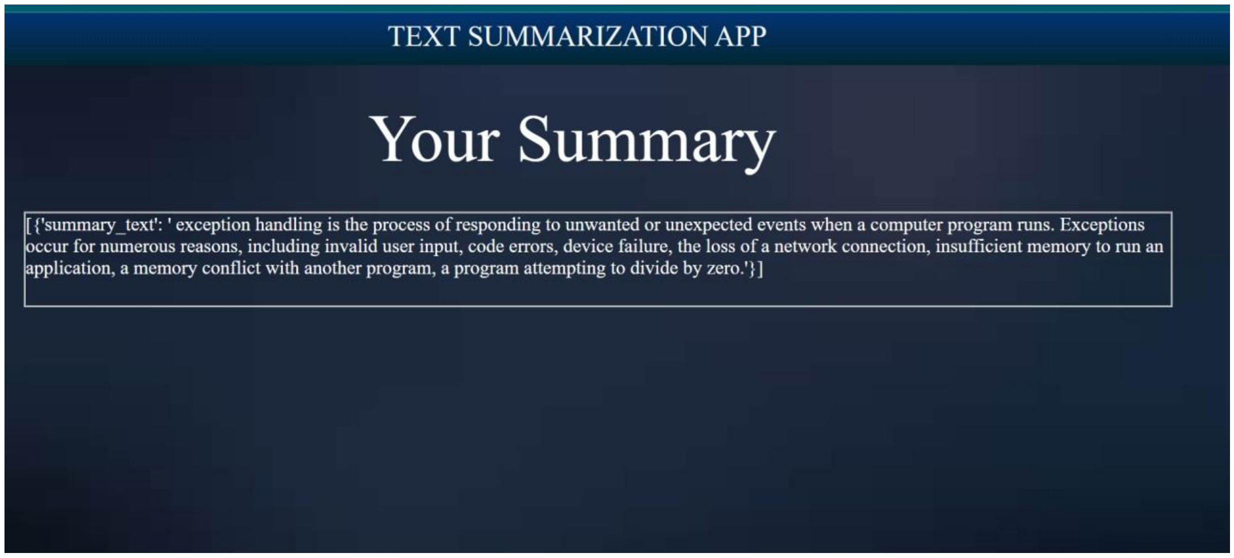
8. OUTPUT SCREENS



```
Exception handling is the process of responding to unwanted or unexpected events when a computer program runs. \\nException handling deals with these
events to avoid the program or system crashing, and without this process, exceptions would disrupt the normal operation of a program. Exceptions occur for
numerous reasons, including invalid user input, code errors, device failure, the loss of a network connection, insufficient memory to run an application, a
memory conflict with another program, a program attempting to divide by zero or a user attempting to open files that are unavailable. When an exception
occurs, specialized programming language constructs, interrupt hardware mechanisms or operating system inter process communication facilities handle the
exception. Exception handling differs from error handling in that the former involves conditions an application might catch versus serious problems an
application might want to avoid. In contrast, error handling helps maintain the normal flow of software program execution
```

Figure:8.1: Home Page

This is the page where users can enter their large document or article which they wants to be summarize.



```
[{'summary_text': ' exception handling is the process of responding to unwanted or unexpected events when a computer program runs. Exceptions
occur for numerous reasons, including invalid user input, code errors, device failure, the loss of a network connection, insufficient memory to run an
application, a memory conflict with another program, a program attempting to divide by zero.'}]
```

Figure 8.2: Summary Generated Page

9. CONCLUSION AND FUTURE SCOPE

9.1 CONCLUSION

In this study, we proposed Hie-BART to can take into account the relationship between words and sentences in BART by dividing the self-attention layer of encoder into word and sentence levels. In the experiments, we confirmed that HieBART improved the F-score of ROUGE-L by 0.23 points relative to the non-hierarchical BART model, and the proposed model was better than the strong baselines, BERTSUM and T5 models for the CNN/Daily Mail dataset. As future work, we intend to investigate methods to incorporate information between sentences in addition to word-to-word and word-to-sentence information.

9.2 FUTURE SCOPE

To develop more accuracy using machine learning algorithms and advanced techniques . The work can be extended and improved for the generation summary for large document using BART model.

10. BIBILIOGRAPHY

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics:

Human Language Technologies, Volume 1 (Long and Short Papers), pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics. Jie Hao, Xing Wang, Shuming Shi, Jinfeng Zhang, and Zhaopeng Tu. 2019. Multi-granularity self-attention for neural machine translation. In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), pages 887–897, Hong Kong, China. Association for Computational Linguistics.

Karl Moritz Hermann, Tomas Kocisky, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching machines to read and comprehend. In Advances in neural information processing systems, pages 1693– 1701. Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre– training for natural language generation, translation

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In Text Summarization Branches Out, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Yang Liu and Mirella Lapata. 2019. text summarization with pretrained encoders. In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), pages 3730–3740, Hong Kong, China. Association for Computational Linguistics.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. Journal of Machine Learning Research, 21(140):1–67.

ABSTRACTIVE TEXT SUMMARIZATION USING BART

P. Gayathri [1], Ch. Dhana Lakshmi [2], M. Naga Sirisha [3] and Sd. Rizwana [4]

[1,2,3] student of Department of Computer Science and Engineering, Narasaraopet Engineering College

Faculty of Department of Computer Science and Engineering, Narasaraopet Engineering College, Narasaraopet

[1]gayathripalepu660@gmail.com , [2] dhanalakshmichelli08@gmail.com, [3]mannepuli123@gmail.com,
[4] syedrizwananrt@gmail.com

Abstract

We have witnessed the rise of automation in recent years for human convenience. With the use of ML learning, we get closer to realizing a general in nature AI. Natural Language Processing, Computer Vision, and Machine Learning are the three main subfields of artificial intelligence. Automated Text Summarization is a key component of Natural Language Processing, which entails the comprehension and manipulation of human language. Text summarizing is reducing a long document to a concise summary. While keeping the information's context (or meaning), it produces information that is fluid and coherent. Making a manual summary is a challenging process for humans because it necessitates a careful examination of the full document.

Keywords: Machine Learning, Text Summarization, BART Model.

1. Introduction

Text summarizing, which involves shrinking the original document's size while retaining its original information, produces a summary that is less than half the size of the original document's main text. One could think about summarization as a two-step procedure. The first stage is to extract important ideas or sequences from the original text by creating an intermediary vector or file. This step may also involve any text pre-processing, such as tokenization, tagging, or other operations, that is necessary. This intermediate file is used to create a summary in the following phase. One example of a text summarizer that enables users to find the news that most interests them is News Blaster. The provocation of writing a concise, pinpoint, and synopsis of a lengthier text document is known as text summarization. In order to better assist in the discovery of relevant information and to consume relevant information more quickly, approaches for automatic text summarizing are urgently needed. Using self-attentions, this technique permits interactions not just between words but also between phrases and words.

The major goal of this project is to lessen the emphasis on giving a trustworthy summary of a lengthy text, which mostly saves people's important time. so that they can concentrate on their best projects. We are employing the BART model of abstractive text summarization to condense this lengthy paper[1]. It greatly aids pupils in learning how to identify key concepts and track down relevant information to bolster those concepts and make them more useful. It aids the kids in developing better concentration abilities so they can concentrate on words and phrases from the lengthy material that has been supplied.

The dataset we have used is CNN Daily Mail Dataset, which contains 3 attributes as id, article, highlights. The id shows the unique id’s for each and every article present in the dataset. And the article attribute contains the long document or article which we need to summarize. And finally highlights attribute contains the summary of the article which can used further in calculating the rouge score passing as reference parameter.

2. Abstractive Text Summarization

Abstractive text summarization is one of the type of text summarization. It evolve newly discovered words and sentences, integrate those in a significant method and then attach the nearly all pivotal particulars in the original document. As a result, abstractive approaches are added arduous to use than extractive summarization strategy and cost more to compute.

As human language is sequential in nature, it has been discovered that RNNs and models based on them perform better than alternative methods. One of the fundamental models for abstractive text summarization is sequence to sequence[2]. One sequence is followed by another (e.g. machine translation). It does this by utilizing DNNs. For extended carry of sequences and to prevent the back propagation issue, this method especially leverages LSTM. The input of one cell becomes the output of the next, and so forth. This enables it to understand the order through a sentence. Encoder-decoder architecture is usually used. The encoder converts an input into a invisible vector that correlate with to it and repress both the item and its value. This concealed vector is transmitted.

3. BART Model

BART is a denoising autoencoder that was trained as a sequence-to-sequence model. This implies that a refined BART model can accept one text sequence as input and output another text sequence. This kind of approach is applicable to text summarization [3], question answering, machine rendering question answering on a specific corpus, question answering, and sequence categorization (labeling input text sentences or tokens). Sentence evocation is other assignment that evaluates whether two or more sentences are reasonably expansion of one another or wisely connected to a given submission.

3.1 Architecture

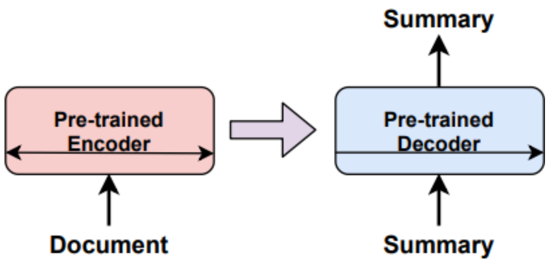


Figure 1: Architecture of BART Model

The building design is very alike to that of Bidirectional Encoder Representation from Transformers, with the backing exceptions:
(1) Every part of the decoder enact cross-attention above the last invisible part of the encoder in addition (as in the transformer sequence-to-sequence model); and (2) Unlike BART[4], Bidirectional Encoder Representation from Transformers employs totalling feed-forward network prior to word prediction. Bidirectional Auto-Regressive Transformers has about ten percent more limitation overall than the Bidirectional Encoder Representation from Transformers model of the same size.

3.2 Pre-training of BART

The cross-entropy connecting the production of the decoder and the real document—the remodeling debt is optimized after manipulated documents in order to train BART [5]. BART enables us to apply any kind of document corruption, unlike existing denoising autoencoders, which are tailored to precise noising plans. Bidirectional Auto-Regressive Transformers is similar to a language model in the inferior case, where all source-associated information is damaged. We test a number of recently developed and creative transformations, but we think there is a lot of room for the creation of additional fresh options. Examples of the transformations we utilized are displayed in Figure 2, and a summary of them is given below.

Token Masking:

Token Masking is used to restore any independent tokens in a sentence. Established the remaining of the succession, the model evolves the potential to forecast the only one token**.**

Sentence Permutation:

Arbitrary permutations are used to whole stops-separated sentences. This support the model's studying of how sentences are wisely implicit.

Document Rotation:

The sequence of the document is interchange to begin with a chosen token[6]. At the edge of the document, the data that came before the token is appended. This supplies particulars on the extensive organization of the document as well as what a document's starting or closing view like.

Token Deletion:

Tokens are eliminated at arbitrary. The model require to have the potential to forecast token data and recognize the position where a token was removed from.

Text Infilling:

Text Infilling places a masked token into a arbitrary chosen position or substitutes word sequences with a only one mask token. Almost precise method is aggregate of sentence permutation and text infilling.

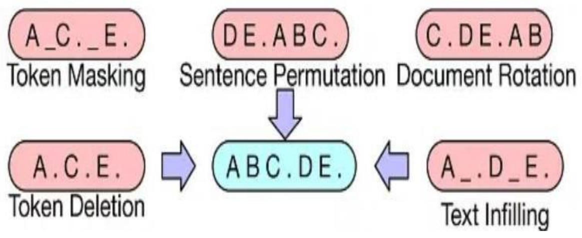
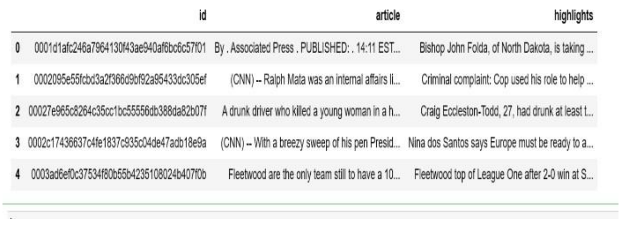


Figure 2: Pre-training Techniques

4. Dataset

The dataset we have used is CNN Daily Mail Dataset, Which contains plenty of news articles int it.



	id	article	highlights
0	0001d1afc246a7964130f43ae940af6bc6c57f01	By . Associated Press . PUBLISHED: . 14:11 EST...	Bishop John Folda, of North Dakota, is taking ...
1	0002095e55fcbd3a2f366d9bf92a95433dc305ef	(CNN) -- Ralph Mata was an internal affairs li...	Criminal complaint: Cop used his role to help ...
2	00027e965c8264c35cc1bc55556db388da82b07f	A drunk driver who killed a young woman in a h...	Craig Eccleston-Todd, 27, had drunk at least t...
3	0002c17436637c4fe1837c935c04de47adb18e9a	(CNN) -- With a breezy sweep of his pen Presid...	Nina dos Santos says Europe must be ready to a...
4	0003ad6ef0c37534f80b55b4235108024b407f0b	Fleetwood are the only team still to have a 10...	Fleetwood top of League One after 2-0 win at S...

Figure 3: Overview of Dataset

The dataset contains 3 attributes "id, article, highlights". The first attribute id provide the unique id for each article present in the dataset. And the second attribute "article" contains long document or news article which we need to summarize. And lastly, the third attribute contains the summary of the each article in their respective highlights column, and this attribute is used as a reference while calculating the rouge score for our machine generated summary.

5. Approaches for Abstractive Text Summarization

5.1 T5 Transformer Model:

An encoder-decoder model is T5. All linguistic issues are transformed into text-to-text formats. You must first import the tokenizer and associated model using the command listed below. T5ForConditionalGeneration model should be used when both the input and output are sequences.

5.2 GPT-2 Model:

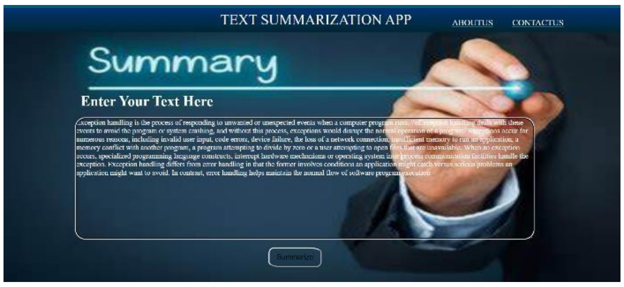
A dataset of 8 million online pages was used to train the 1.5 billion parameter language model GPT-2. The consecutive leading objective of GPT-2's training is to anticipate the successive word from the text's previous words. This straight forward aim incorporate absolute examples of plentiful jobs from distinct territory due to the diversity of the dataset. With more than 10X the parameters and trained on more than 10X the quantity of data, GPT-2 is a straight scale-up of GPT. As seen in the sample script run generation.py, this capability enables GPT-2 to produce syntactically consistent text. The previously computed key/value attention pairs can be used as input for the PyTorch models. By using this historical value, the model is prevented from recalculating previously computed values when text is generated.

5.3 Pegasus Model:

PEGASUS trains a machine to anticipate sentences by removing whole sentences from a document and the model is trained to forecast the sentences. It constructs logical to pre-train the encoder as a masked language model as PEGASUS's base building design contains of an encoder and a decoder even if the GSG is its key addition. A Modern Approach to Abstractive Text Summarization Reading a document and writing a summary is a common assignment for students.

6. Results

In the first page of application, there will be a text area where users who wants the summary of a long document will enter their text .



TEXT SUMMARIZATION APP ABOUTUS CONTACTUS

Summary

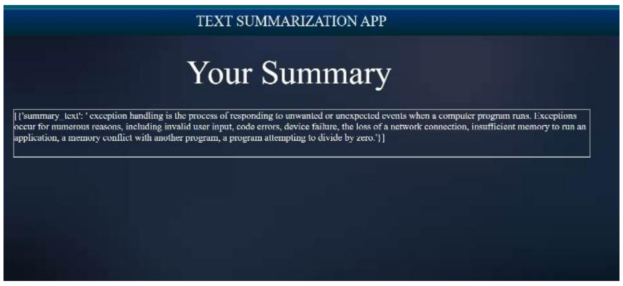
Enter Your Text Here

Exception handling is the process of responding to unwanted or unexpected events when a computer program runs. [illegible] exception handling deals with these events to avoid the program or system crashing, and without this process, exceptions would disrupt the normal operation of a program. Exceptions occur for numerous reasons, including invalid user input, code errors, device failure, the loss of a network connection, insufficient memory to run an application, a memory conflict with another program, a program attempting to divide by zero or a user attempting to open files that are unavailable. When an exception occurs, specialized programming language constructs, interrupt hardware mechanisms or operating system inter-process communication facilities handle the exception. Exception handling differs from error handling in that the former involves conditions an application might catch versus serious problems an application might want to avoid. In contrast, error handling helps maintain the normal flow of software program execution.

Summarize

Figure 4: Text Entering Page

And then after clicking on summarize button, it will navigate to another page. Where the machine generated summary will be displayed.



TEXT SUMMARIZATION APP

Your Summary

[{'summary_text': ' exception handling is the process of responding to unwanted or unexpected events when a computer program runs. Exceptions occur for numerous reasons, including invalid user input, code errors, device failure, the loss of a network connection, insufficient memory to run an application, a memory conflict with another program, a program attempting to divide by zero.'}]

Fig:5 Summary Generated Page

6.1 Result Analysis:

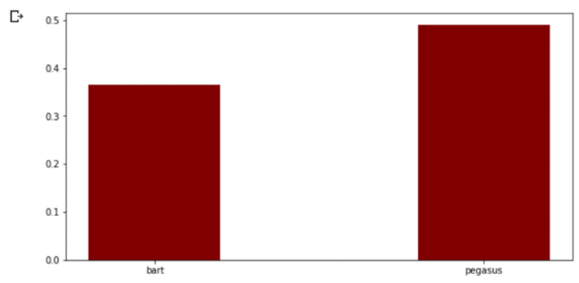


Figure:6 Comparison of Models

7. CONCLUSION

Automatic text summarization is a difficult problem, with attempts and solutions dating back a few years. as a result of the release of strong hardware and the as a result of the rise in computing power, we may use more complicated algorithms to get the desired outcome. The statistical (but computationally less expensive) extractive summarizing techniques are thus gradually giving way to abstractive summarization techniques using non-linear models (deep learning). While still a challenging undertaking, creating an abstract using the abstractive summarization method. Abstractive summarizing techniques result in more comprehensive, coherent, and information-rich summaries. The study of abstractive summarization approaches is more beneficial for the aforementioned causes. And we have evaluated the resulted summary with ROUGE [7].

To improve accuracy through the use of cutting-edge methods and machine learning algorithms. The work for using the BART model to generate a summary for a lengthy document can be enhanced and expanded.

8. REFERENCES

[1] Ramesh Nallapati, Feifei Zhai, and Bowen Zhou. 2017. SummaRuNNer: A recurrent neural network based sequence model for extractive. In Proceedings of In Association for the Advancement of Artificial Intelligence.

[2] Yang Liu and Mirella Lapata. 2019. Text summarization with pretrained encoders. In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), pages 3730–3740, Hong Kong, China. Association for Computational Linguistics.

[3] Jie Hao, Xing Wang, Shuming Shi, Jinfeng Zhang, and Zhaopeng Tu. 2019. Multi-granularity self-attention for neural machine translation. In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), pages 887–897, Hong Kong, China. Association for Computational Linguistics.

[4] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. Journal of Machine Learning Research, 21(140):1–67.

[5] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

[6] Karl Moritz Hermann, Tomas Kocisky, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching machines to read and comprehend. In Advances in neural information processing systems, pages 1693– 1701.

[7] Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In Text Summarization Branches Out, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

[8] Abigail See, Peter J. Liu, and Christopher D. Manning. 2017. Get to the point: Summarization with pointer-generator networks. In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 1073–1083, Vancouver, Canada. Association for Computational Linguistics.

[9] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In Advances in Neural Information Processing Systems, volume 30, pages 5998–6008. Curran Associates, Inc

[10] S. Rafi and R. Das, "RNN Encoder And Decoder With Teacher Forcing Attention Mechanism for Abstractive Summarization," *2021 IEEE 18th India Council International Conference (INDICON)*, Guwahati, India, 2021, pp. 1-7, doi: 10.1109/INDICON52576.2021.9691681.

[11] Debnath, D., Das, R., Rafi, S. (2022). Sentiment-Based Abstractive Text Summarization Using Attention Oriented LSTM Model. In: Satapathy, S.C., Peer, P., Tang, J., Bhateja, V., Ghosh, A. (eds) Intelligent Data Engineering and Analytics. Smart Innovation, Systems and Technologies, vol 266. Springer, Singapore. https://doi.org/10.1007/978-981-16-6624-7_20.

[12] S. Rafi and R. Das, "A Linear Sub-Structure with Co-Variance Shift for Image Captioning," *2021 8th International Conference on Soft Computing & Machine Intelligence (ISCMI)*, Cario, Egypt, 2021, pp. 242-246, doi: 10.1109/ISCMI53840.2021.9654828.

DG7-2

ORIGINALITY REPORT

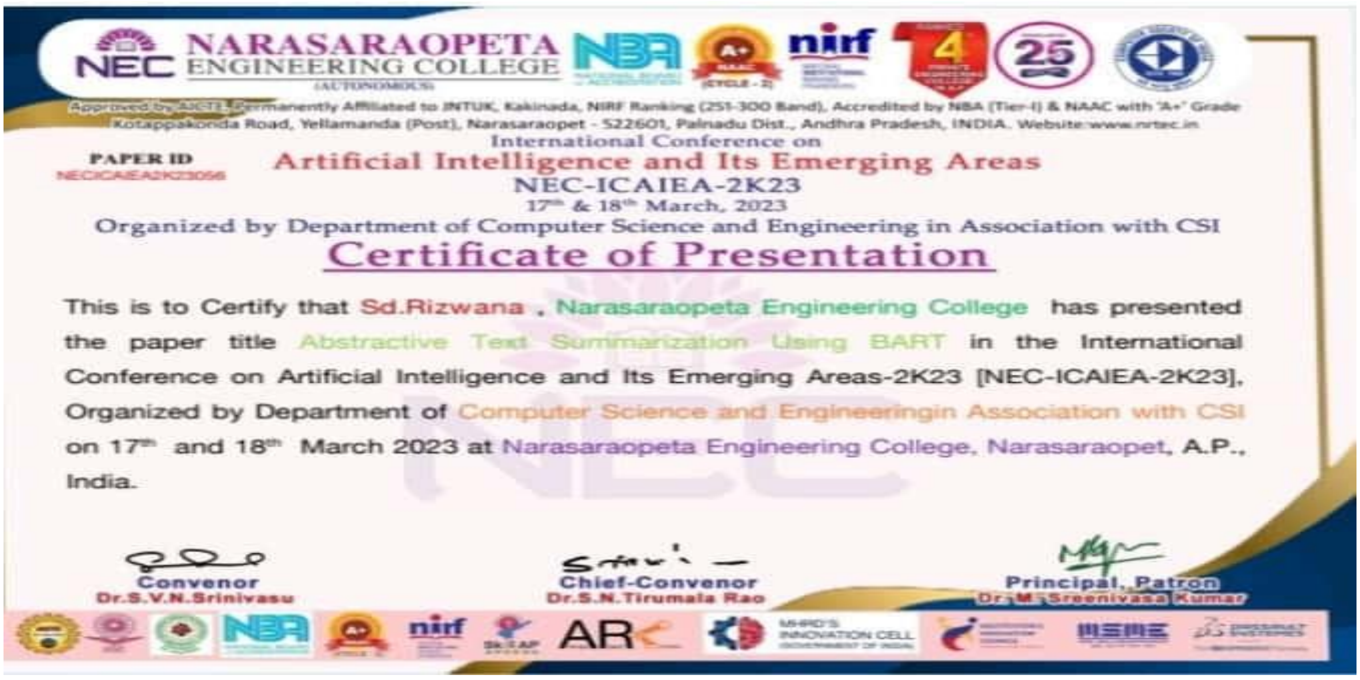
11%	4%	4%	4%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to Liverpool John Moores University Student Paper	3%
2	"Natural Language Processing and Chinese Computing", Springer Science and Business Media LLC, 2018 Publication	1%
3	Submitted to University of Adelaide Student Paper	1%
4	Mark Amerika. "1 ONTO- OPERATIONAL PRESENCE Artificial Creative Intelligence as Meta Remix Engine", Walter de Gruyter GmbH, 2022 Publication	1%
5	www.ijcseonline.isroset.org Internet Source	1%
6	nasscom.in Internet Source	1%
7	aclanthology.org Internet Source	1%

8	www.mitpressjournals.org Internet Source	1%
9	medium.com Internet Source	<1%
10	www.jetir.org Internet Source	<1%
11	www.statmt.org Internet Source	<1%
12	"A Pointer Generator Network Model to Automatic Text Summarization and Headline Generation", International Journal of Engineering and Advanced Technology, 2019 Publication	<1%
13	Tian Shi, Yaser Keneshloo, Naren Ramakrishnan, Chandan K. Reddy. "Neural Abstractive Text Summarization with Sequence-to-Sequence Models", ACM/IMS Transactions on Data Science, 2021 Publication	<1%

Exclude quotes	On	Exclude matches	Off
Exclude bibliography	On		



NEC NARASARAOPETA ENGINEERING COLLEGE
(AUTONOMOUS)

Approved by AICTE, Permanently Affiliated to JNTUK, Kakinada, NIRF Ranking (251-300 Band), Accredited by NBA (Tier-I) & NAAC with 'A+' Grade
Kotappakonda Road, Yellamanda (Post), Narasaraopet - 522601, Palnadu Dist., Andhra Pradesh, INDIA. Website:www.nrtec.in

International Conference on

PAPER ID
NECICAIEA2K23056

Artificial Intelligence and Its Emerging Areas

NEC-ICAIEA-2K23

17th & 18th March, 2023

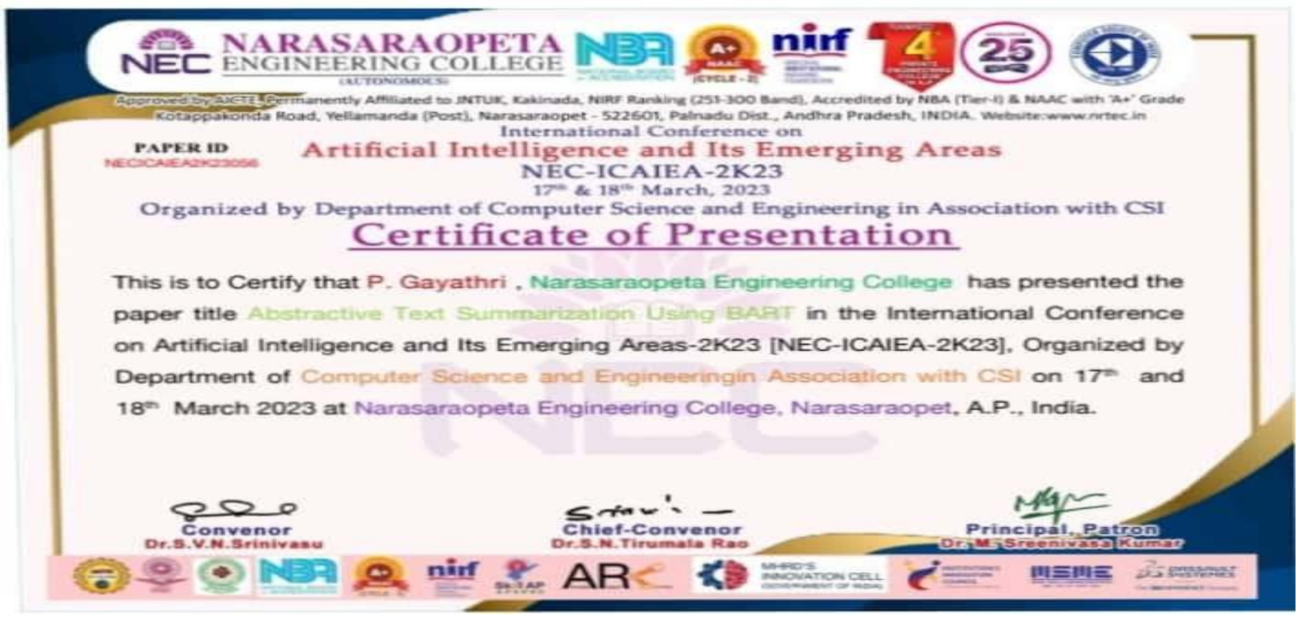
Organized by Department of Computer Science and Engineering in Association with CSI

Certificate of Presentation

This is to Certify that Sd.Rizwana , Narasaraopeta Engineering College has presented the paper title Abstractive Text Summarization Using BART in the International Conference on Artificial Intelligence and Its Emerging Areas-2K23 [NEC-ICAIEA-2K23], Organized by Department of Computer Science and Engineeringin Association with CSI on 17th and 18th March 2023 at Narasaraopeta Engineering College, Narasaraopet, A.P., India.

[illegible]	[illegible]	[illegible]
Convenor	Chief-Convenor	Principal, Patron
Dr.S.V.N.Srinivasu	Dr.S.N.Tirumala Rao	Dr.M.Sreenivasa Kumar

MHRD'S INNOVATION CELL
GOVERNMENT OF INDIA



NEC NARASARAOPETA ENGINEERING COLLEGE (AUTONOMOUS)

Approved by AICTE, Permanently Affiliated to JNTUK, Kakinada, NIRF Ranking (251-300 Band), Accredited by NBA (Tier-I) & NAAC with 'A+' Grade
Kotappakonda Road, Yellamanda (Post), Narasaraopet - 522601, Palnadu Dist., Andhra Pradesh, INDIA. Website:www.nrtec.in

International Conference on

PAPER ID
NECICAIEA2K23056

Artificial Intelligence and Its Emerging Areas
NEC-ICAIEA-2K23
17th & 18th March, 2023

Organized by Department of Computer Science and Engineering in Association with CSI

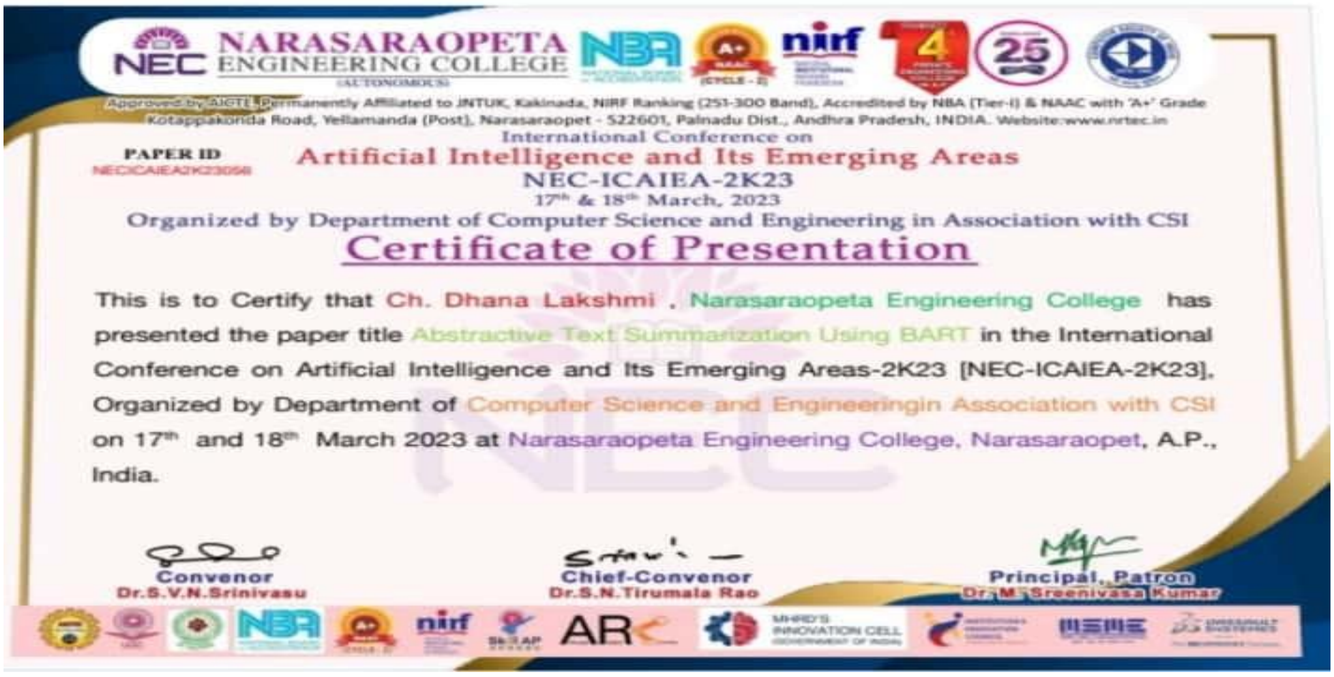
Certificate of Presentation

This is to Certify that P. Gayathri , Narasaraopeta Engineering College has presented the paper title Abstractive Text Summarization Using BART in the International Conference on Artificial Intelligence and Its Emerging Areas-2K23 [NEC-ICAIEA-2K23], Organized by Department of Computer Science and Engineeringin Association with CSI on 17th and 18th March 2023 at Narasaraopeta Engineering College, Narasaraopet, A.P., India.

[illegible]
Convenor
Dr.S.V.N.Srinivasu

[illegible]
Chief-Convenor
Dr.S.N.Tirumala Rao

[illegible]
Principal, Patron
Dr. M. Sreenivasa Kumar



NEC NARASARAOPETA ENGINEERING COLLEGE (AUTONOMOUS)

Approved by AICTE, Permanently Affiliated to JNTUK, Kakinada, NIRF Ranking (251-300 Band), Accredited by NBA (Tier-I) & NAAC with 'A+' Grade
Kotappakonda Road, Yellamanda (Post), Narasaraopet - 522601, Palnadu Dist., Andhra Pradesh, INDIA. Website:www.nrtec.in

PAPER ID
NECICAIEA2K23056

International Conference on
Artificial Intelligence and Its Emerging Areas
NEC-ICAIEA-2K23
17th & 18th March, 2023
Organized by Department of Computer Science and Engineering in Association with CSI

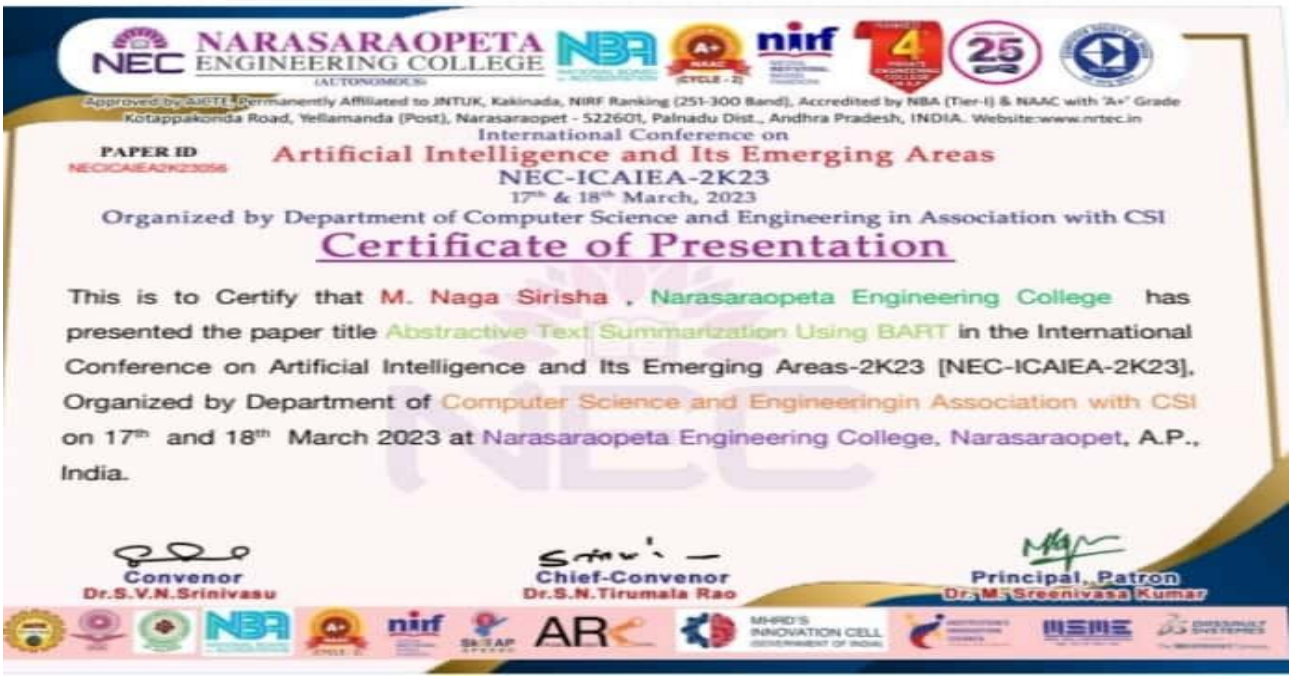
Certificate of Presentation

This is to Certify that Ch. Dhana Lakshmi , Narasaraopeta Engineering College has presented the paper title Abstractive Text Summarization Using BART in the International Conference on Artificial Intelligence and Its Emerging Areas-2K23 [NEC-ICAIEA-2K23], Organized by Department of Computer Science and Engineeringin Association with CSI on 17th and 18th March 2023 at Narasaraopeta Engineering College, Narasaraopet, A.P., India.

[illegible]
Convenor
Dr.S.V.N.Srinivasu

[illegible]
Chief-Convenor
Dr.S.N.Tirumala Rao

[illegible]
Principal, Patron
Dr.M.Sreenivasa Kumar



NARASAROPETA ENGINEERING COLLEGE (AUTONOMOUS)

Approved by AICTE, Permanently Affiliated to JNTUK, Kakinada, NIRF Ranking (251-300 Band), Accredited by NBA (Tier-I) & NAAC with 'A+' Grade
Kotappakonda Road, Yellamanda (Post), Narasaraopet - 522601, Palnadu Dist., Andhra Pradesh, INDIA. Website:www.nrtec.in

PAPER ID
NECICAIEA2K23056

International Conference on
Artificial Intelligence and Its Emerging Areas
NEC-ICAIEA-2K23
17th & 18th March, 2023
Organized by Department of Computer Science and Engineering in Association with CSI

Certificate of Presentation

This is to Certify that M. Naga Sirisha , Narasaraopeta Engineering College has presented the paper title Abstractive Text Summarization Using BART in the International Conference on Artificial Intelligence and Its Emerging Areas-2K23 [NEC-ICAIEA-2K23], Organized by Department of Computer Science and Engineeringin Association with CSI on 17th and 18th March 2023 at Narasaraopeta Engineering College, Narasaraopet, A.P., India.

[illegible]	[illegible]	[illegible]
Convenor	Chief-Convenor	Principal, Patron
Dr.S.V.N.Srinivasu	Dr.S.N.Tirumala Rao	Dr. M. Sreenivasa Kumar

MHRD'S INNOVATION CELL GOVERNMENT OF INDIA