DHANALAKSHMI SRINIVASAN

COLLEGE OF ENGINEERING AND TECHNOLOGY
MAMALLAPURAM, CHENNAI - 603 104.

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

A MINI PROJECT REPORT

EFFECTIVE DEEP LEARNING BASED ABSTRACTIVE TEXT SUMMARISATION AND IMAGE CAPTIONING

Submitted by

MOHANRAJ K	**310520104073**
PURUSHOTHAMAN N	**310520104088**
RAGHUL PRASAD S	**301520104090**
THIRUVENKADAM B	**310520104132**

in partial fulfilment for the award of the degree

of

BACHELOR OF ENGINEERING

In

COMPUTER SCIENCE AND ENGINEERING

ANNA UNIVERSITY :: CHENNAI - 600 025

MAY 2023

ANNA UNIVERSITY :: CHENNAI – 600 025

BONAFIDE CERTIFICATE

A Certificate that, this Mini project report **"EFFECTIVE DEEP LEARNING BASED ABSTRACTIVE TEXT SUMMERIZATION AND IMAGE CAPTIONING"** is the bonafide work of "**MOHANRAJ K(310520104073**), **PURUSHOTHAMAN N(310520104088), RAGHUL PRASAD S (310520104090**), **THIRUVENKADAM B(310520104132)"** who carried out the Mini project work under my supervision.

SIGNATURE

HEAD OF THE DEPARTMENT
Dr.P.Malathi,PHD
Department of Computer Science and Engineering,
Dhanalakshmi Srinivasan College of Engineering & Technology,
Mahabalipuram.

SIGNATURE

SUPERVISOR
Mr.Senthil,M.E
Department of Computer Science and Engineering,
Dhanalakshmi Srinivasan College of Engineering & Technology,
Mahabalipuram.

Submitted for the Practical Examination held on _____

INTERNAL EXAMINER

EXTERNAL EXAMINER

ABSTRACT

The technique of text summarizing turns a lengthy text into a summary. The foundation of earlier information retrieval and summarization algorithms is a large labelled dataset that uses manually created characteristics. For a specific domain and focused on the specific sub-domain to increase efficiency. This research introduces a novel text summarization model for deep learning (DL) based information retrieval. The three main processes that make up the proposed model are text summarization, template development, and information retrieval. For recovering textual material at first, the bidirectional long short term memory (BiLSTM) technique is used, which assumes each word d, takes the data from the sentence, and embeds it in the semantic vector. Following that, the DL model is used to generate templates. The text is summarized using the deep belief network (DBN) model as a text summary technique. Also, the visible entities that are present in the photographs are given their own image descriptions. The design of BiLSTM with the DBN model for the text summarization and image captioning process shows the novelty of the work. Giga and DUC corpora are used to validate the performance of the proposed technique.

ACKNOWLEDGEMENT

First of all, we thank, **Our Almighty** for his blessings upon us to strengthen our minds and soul to take up this project. We owe a great many thanks to a great many people who helped and supported us in this project.

We thank Our **Chairman, THIRU. A. SRINIVASAN**, who allowed us to do the project.

We are also thankful to Our **Principal**, **Dr. R. SARAVANAN, (Ph.D.)** for his constant support to do the project.

We extremely thank Our **Vice Principal**, **Dr. V. JANAKIRAMAN, (Ph. D.),** for his constant support in selecting the project.

We are grateful to Our **Head of the department**, **Dr. P. MALATHI**, (**Ph.D.),** who expressed her interest and guided our work, and supplied us with some useful ideas.

We would like to thank our Guide **Assistant Professor Mr. SENTHIL**, for following our project with interest and for giving me constant support. He taught us not only how to do the project, but also how to enjoy the project.

We wish to extend our grateful acknowledgment and sincere thanks to our project coordinators, Dr. P. Malathi and Mr. Prabakaran MV for their constant encouragement and kind support in completing the project

Furthermore, we would like to thank all our **Teaching Faculty and Non- teaching Faculty** for their timely help in solving any project queries.

Finally, we would like to thank our **Parents** for their blessings, support, and encouragement throughout our life.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF ABBREVATIONS

S.NO	ABBREVAATION	EXPANSION
1	CNN	CONVOLUTIONAL NEURAL NETWORK
2	RNN	RECURRENT NEURAL NETWORK
3	ANN	ARTIFICIAL NEURAL NETWORK
4	ROUGE	RECALL-ORIENTED UNDERSTRUDY FOR GISTING EVALUVATION
5	RAM	RANDOM ACCESS MEMORY
6	CPU	CENTRAL PROCESSING UNIT
7	GPU	GRAPHICS PROCESSING UNIT
8	IDE	INTEGRATED DEVELOPMENT ENVIRONMENT

CHAPTER 1

INTRODUCTION

1.1. DOMAIN INTRODUCTION

Deep learning is a branch of machine learning which is completely based on artificial neural networks, as neural network is going to mimic the human brain so deep learning is also a kind of mimic of human brain. In deep learning, we don't need to explicitly program everything. The concept of deep learning is not new. It has been around for a couple of years now. Deep Learning models are able to automatically learn features from the data, which makes them well-suited for tasks such as image recognition, speech recognition, and natural language processing. The most widely used architectures in deep learning are feedforward neural networks, convolutional neural networks (CNNs), and recurrent neural networks (RNNs). In human brain approximately 100 billion neurons all together this is a picture of an individual neuron and each neuron is connected through thousand of their neighbours. So, we create an artificial structure called an artificial neural net where we have nodes or neurons. We have some neurons for input value and some for output value and in between, there may be lots of neurons interconnected in the hidden layer.

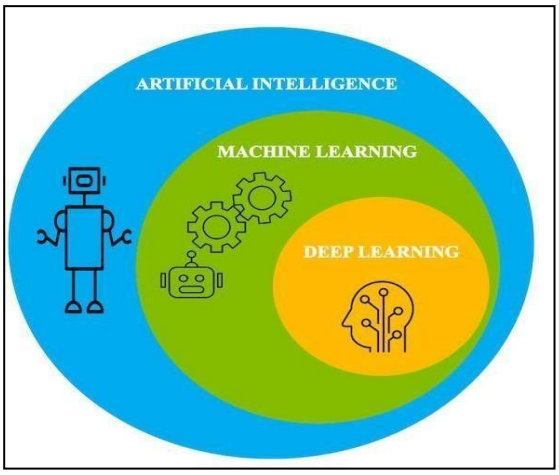

FIGURE 1.1 CLASSIFICATION OF DOMAIN

1.2. FUNDAMENTALS OF DEEP LEARNING

Deep learning is a subfield of machine learning that involves training artificial neural networks to solve complex problems, often with large amounts of data. Artificial neural networks are computational models that are inspired by the structure and function of biological neurons in the brain. Neural networks consist of layers of interconnected nodes, where each node represents a mathematical function that takes input from the nodes in the previous layer and produces output for the nodes in the next layer. Backpropagation is a mathematical algorithm used to train neural networks by iteratively adjusting the weights and biases of the nodes in the network to minimize the difference between the predicted outputs and the actual outputs.Activation functions are mathematical functions that are applied to the output of each node in a neural network to introduce nonlinearity into the model. Common activation functions include sigmoid, ReLU Convolutional neural networks (CNNs) are a type of neural network that are specialized for image and video recognition tasks.

CNNs use convolutional layers to extract spatial features from input images, and pooling layers to down sample the feature maps. Recurrent neural networks (RNNs) are a type of neural network that are specialized for sequence data, such as time series or natural language processing tasks. RNNs use recurrent layers to maintain a state that represents the context of previous inputs, which allows the model to learn temporal dependencies. Transfer learning is a technique where a pre-trained neural network is used as a starting point for a new task, rather than training a new network from scratch. Transfer learning can significantly reduce the amount of data and computing resources needed to train a new model. Deep learning is a class of machine learning algorithms that uses multiple layers to progressively extract higher-level features from the raw input. For example, in image processing, lower layers may identify edges, while

higher layers may identify the concepts relevant to a human such as digits or letters or faces.

1.3. COMPONENTS OF DEEP LEARNING

The main components of deep learning include: Artificial Neural Networks (ANNs) as mentioned earlier, ANNs are the computational models that are inspired by the structure and function of biological neurons in the brain. ANNs consist of interconnected nodes or neurons, where each node takes input from the previous layer and produces output for the next layer. ANNs can have several hidden layers, hence the term "deep" learning. Activation functions introduce nonlinearity into the output of each neuron in a neural network. This nonlinearity is essential for the network to learn complex functions. Popular activation functions include sigmoid, ReLU, and tanh. Backpropagation is a mathematical algorithm used to adjust the weights and biases of the neurons in a neural network to minimize the difference between the predicted outputs and the actual outputs. It is based on the chain rule of calculus and allows us to compute the gradient of the loss function with respect to the weights and biases of the network. loss function measures how well the neural network is performing on a particular task. The goal of deep learning is to minimize the loss function by adjusting the weights and biases of the network using backpropagation.

Optimization algorithms, such as stochastic gradient descent (SGD), Adam, and RMSprop, are used to update the weights and biases of the network during training. These algorithms use the gradient computed by backpropagation to adjust the network parameters. Regularization techniques, such as L1/L2 regularization and dropout, are used to prevent overfitting in a neural network. Overfitting occurs when the network becomes too complex and starts to fit noise in the training data rather than the underlying pattern. CNNs are a type of neural network that are specialized for image and video recognition tasks. They use

convolutional layers to extract spatial features from input images, and pooling layers to down sample the feature maps. Recurrent Neural Networks (RNNs) are a type of neural network that are specialized for sequence data.

1.4. PROBLEM DEFINITION

Text summarization and image captioning are two different applications of natural language processing and computer vision, respectively. However, in both cases, problem definition plays a crucial role in determining the effectiveness of the solution. In text summarization, the problem definition involves identifying the key information in a given text and presenting it in a concise and coherent manner. This requires understanding the context of the text and the intended audience, as well as determining the importance of different sentences and phrases. The goal is to provide a summary that captures the main ideas and key points of the original text, while also being readable and informative.

In image captioning, the problem definition involves generating a descriptive sentence or phrase that accurately conveys the content of an image. This requires understanding the objects, people, and activities depicted in the image, as well as the overall composition and context. The goal is to provide a caption that is informative, accurate, and engaging, while also being concise and grammatically correct. In both cases, the problem definition is critical to developing effective solutions that meet the needs of users and stakeholders. This requires careful analysis of the problem domain, identification of relevant data sources and algorithms, and evaluation of the effectiveness of the solution. The problem definition in text summarization for a report involves identifying the most important information from the report and presenting it in a condensed form that highlights key takeaways. The challenge is to accurately represent the content of the report while avoiding unnecessary details that may distract from the main message. In image captioning for a report, the problem definition involves generating captions that describe the images included in the report. The captions

should be informative and relevant to the content of the report, providing additional context or highlighting key points.

1.5. PROJECT DESCRIPTION

Text summarization and image captioning are two separate but related fields of research in artificial intelligence that aim to automatically generate concise and descriptive representations of text and images, respectively. Text summarization involves the task of generating a brief summary of a longer text document, such as a news article or a scientific paper. This can be done using different techniques, including extractive and abstractive summarization. Extractive summarization involves selecting the most important sentences or phrases from the original text and using them to form a summary. Abstractive summarization, on the other hand, involves generating new sentences that capture the main ideas of the original text. Image captioning, on the other hand, involves the task of generating a descriptive sentence or phrase that summarizes the content of an image.

This can be done using deep learning techniques that analyse the visual features of the image and generate a natural language sentence that describes what is happening in the image. Both text summarization and image captioning have numerous applications in a variety of fields, including journalism, content creation, and information retrieval.

CHAPTER 2

LITERATURE SURVEY

[1] Kanika Agrawal et.al., In this advanced universe of quick progress, today we are having a gigantic measure of information in our storehouses. Since the information is enormous it needs the system by which the information can be proficiently investigated in a short period of time. Henceforth it is the text mining methods which aides in doing as such. It utilizes a few sorts of examples for information retrieval which aides in further examination. The determination of right strategy helps in diminishing the time and expands the throughput. The process of removing extraneous content from the document while retaining the major points of the text in order to shorten the length of the paragraph is called text summarization. This research focuses on to help various lawyers to help in their cases

[2]ZohairMalki et.al., Fall detection comes in the domain of human activity recognition in simple or complex environments and the source of data is mostly from sensors. The literature so till dates are dominated with human crafted features that enable to handle complex actions based on high dimensional sensordatasets. Nowadays, the direction of such research is shifted towards deeplearning executing on high computational power machines with potential applications in health care systems such as assisting elderly people. As the neural networks become deeper and deeper, it becomes challenging to train them. To address this challenge, the residual neural networks come in place as they train by jumping certain layers during training instead of sequentially He et.al., 2016. Ones health condition highly affects the action to be performed. Studying and

evaluating human activities in a systematic way enables to study the behaviour ofindividuals and such study can have several applications such as in health care systems, in surveillance and security Elbowoods et.al., 2019.

[3]Deepali Jain et.al., The advancements in web has led to the easy availability of online legal documents, which are quite long and have complex structures, which is why it is difficult for legal practitioners to go through the document completely. In order to get a quick understanding of these documents, expert summarization is required which is both costly and time consuming. Whereas if effective text summarization can be done automatically, then most of these costly and time consuming options can be avoided. This necessity of automatic techniques has led to the development of the legal document summarization field. In the literature, several works have been found which deals with legal extractive text summarization are 10, 1416, 18, 22, 23, 25, 27, 28, 38, 42. There are mainly two types of text summarization abstractive and extractive. Abstractive summarization deals with producing summaries by generating new texts, whereas extractive summarization builds a summary by selecting the most informative sentences from the document itself.

[4]P.Mahalakshmi et.al., Information retrieval IR defines the process of exploringand acquiring specific data resources which are related to certain details of the resources pool. It is generally found in several applications like web exploration, electronic libraries, digital health records, etc. Since the data resources producedafter searching might be larger in quantity and varied in supremacy, it is significant to grade the data for determining the

degree of relevance. The rankingmodel differentiates IR issues from alternate problems. Hence, ranking approaches are considered as major component of IR studies. Generally, users ofer a request to search engine along with a query, which might be of keywords,numbers and alphabets.

[5]Junzhong Ji et.al., Image captioning, which aims to describe the semantic content of an image in natural language automatically, is a challenging problem in the artificial intelligence community. An increasing number of works focus on this task, which is stimulated by a variety of meaningful practical applications Manuscript received September 5, 2019 revised February 20, 2020 and May 10, 2020 accepted June 9, 2020. Date of publication June 30, 2020 date of current version July 13, 2020. This work was supported in part by the National Natural Science Foundation of China under Grant 61906007, Grant 61672065, Grant 61906011, and Grant 61902378, and in part by the Beijing Municipal Science and Technology Project under Grant KM202010005014. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Ming-Ming Cheng.

[6]Ming-Hsiang Su et.al., The work was supported in part by the Ministry of Science and Technology, Taiwan, under Contract No. MOST 1082221-E-006- 103-MY3. M. -H. Su is with the Department of Computer Science and Information Engineering, National Cheng Kung University e-mail huntfox. sugmail. com. the problem of how to quickly understand these large amounts of information in a limited time. Unlike the quality of articles that are strictly controlled in Wikipedia, content farm websites quickly grow with low-quality articles for high click-through rates. For

example, Demand Media, a content farmwebsite, published one million articles per month in 2009, four times as many asEnglish Wikipedia 2. Content farms have even caused the proliferation of fake news 3.

[7]Ayesha ayub syed et.al., Specifically abstractive summarization based on neural networks. The proposed conceptual framework includes five key elements identified as encoder-decoder architecture, mechanisms, training strategies and optimization algorithms, dataset, and evaluation. Text summarization is an important tool for summarizing text documents in the scientific literature. It can be used for search engine snippets, news websites, lawsuit abstraction, biomedical and clinical text summarization, and other applications. Automatic summarization systems can be modeled in two ways, either using an extractive approach or an abstractive approach. Extractive techniques extract the main sections of the text and concatenate them to produce a summary, while abstractivetechniques paraphrase the text to generate a summary with words different from the original text.

[8] M.f.mridha et.al., The amount of textual material on the web and other libraries is growing tremendously daily. Information utilization has become an expensive and time-consuming activity since data expands in a large quantity at a time and includes irrelevant content or noise. Text summarization is a method used to summarize the data. A manual text summarization process is undoubtedly an effective way to preserve the meaning of the text however, this is a time- consuming activity. Another approach is to utilize the automatic text summarization ATS.

[9]jeongwan shin et.al., Abstractive summarization aims at generating a short andconcise summary from a long source text where the summary captures key information of the source text. Recent abstractive summarization models have shown high performance in automatic evaluation metrics such as ROUGE with the help of The associate editor coordinating the review of this manuscript and approving it for publication was Wai-Keung Fung . pre-trained language models 1, 2, 3. Despite their high performance, they often generate a factually- inconsistent summary with respect to a source text. According to the previous studies 4, 5, 6, 7, about 30 of summaries generated by abstractive summarizationmodels contain factual errors.

[10]Jeongwan et.al., shin In natural language processing, text summarization is animportant application used to extract desired information by reducing large text. Existing studies use keyword-based algorithms for grouping text, which do not give the documents actual theme. Our proposed dynamic corpus creation mechanism combines metadata with summarized extracted text. The proposed approach analyzes the mesh of multiple unstructured documents and generates a linked set of multiple weighted nodes by applying multistage Clustering.

CHAPTER 3

SYSTEM REQUIREMENTS

3.1. HARDWARE REQUIREMENTS

The hardware requirements for text summarization and image captioning can vary depending on the complexity of the task and the size of the dataset being processed. However, in general, the hardware requirements for text summarization are not particularly demanding and can be met by most modern computers. Here are some general hardware recommendations for text summarization:

CPU: A multi-core CPU is recommended for faster processing times, especially when dealing with large datasets.

RAM: The amount of RAM needed will depend on the size of the dataset being processed. As a general rule, it is recommended to have at least 8GB of RAM available for text summarization tasks.

Storage: Sufficient storage space is required to store the input data, the generated summary, and any intermediate files created during the summarization process.

GPU (optional): If you plan to use deep learning models for text summarization, a dedicated GPU can significantly speed up the training process. However, this is not strictly necessary for smaller datasets or less complex summarization tasks.

In summary, for most text summarization tasks, a modern computer with a multi-core CPU, at least 8GB of RAM, and sufficient storage space should be sufficient. If you plan to use deep learning models, a dedicated GPU can also be beneficial.

3.2. SOFTWARE REQUIREMENTS

The software requirements for text summarization and image captioning can vary depending on the specific method or algorithm being used for the task. However, there are some general software requirements that apply to most approaches.

Programming language: Both text summarization and image captioning can be implemented using Python .Python is a popular choice due to its ease of use and the availability of many natural language processing and computer vision libraries.

Natural Language Processing (NLP) and Computer Vision (CV) libraries: Depending on the method used for text summarization and image captioning, various NLP and CV libraries may be required. Popular libraries for NLP include NLTK, Spacey, and Genism, while popular libraries for CV include OpenCV, TensorFlow, and Py Torch.

Deep learning frameworks: Many advanced approaches to text summarization and image captioning involve the use of deep learning models. Therefore, deep learning frameworks such as TensorFlow, Keras, and PyTorch may be required to build and train these models.

Development environment: A development environment such as Jupyter Notebook or an Integrated Development Environment (IDE) such as Google Collab can be helpful for coding, testing, and debugging text summarization and image captioning algorithms.

Version control: It is important to use version control software such as Git to manage code changes and collaborate with other developers working on the project.

CHAPTER 4

SYSTEM ANALYSIS

4.1. EXISTING SYSTEM

The working process involved in the presented DL based information retrieval and text summarization processes which comprises three major stages. Initially, Bi-LSTM based information retrieval and template generation take place. Then, the DBN model is used for text summarization. In addition, CNN and RNN technique facilitates in generating captions to the images. These processes are discussed in the subsequent sections. The dropout is deployed to precede regularization operation. The working process involved in the presented DL based information retrieval and text summarization processes is which comprises three major stages. Initially, Bi-LSTM based information retrieval and template generation take place. Then, the DBN model is used for text summarization. In addition, CNN (Convolutional Neural Network) and RNN (recurrent neural network) technique facilitates in generating captions to the images. The BiLSTM approach is employed to retrieve the textual data, which assumes each word in a sentence extracts the information and embeds it into the semantic vector. Subsequently, the template generation process takes place using the DL model. DBN model is employed as a text summarization tool to summarize the textual content and the image captions are generated.

4.1.1. DRAWBACKS OF EXISTING SYSTEM

CNNs which include the fact that a lot of training data isneeded for the CNN to be effective and that they fail to encode the position and orientation of objects.

They fail to encode the position and orientation of objects.They have a hard time classifying images with different positions. Gradient exploding and vanishing problems are major issues in RNNs.

4.2. PROPOSED SYSTEM

To overcome the lack of quality and accuracy in text summarization and image captioning using transformers model .By using transformers model it achieve very high translate quality even after being trained only for shorter period of time. It works in a inherently sequential in nature and allow for much more parallelization than sequential models. There are three important features in transformers they are, Positional encoding.(describes the location or position of an entity in a sequence so that each position is assigned a unique representation) Attention.(helps to draw connections between any parts of the sequence,) Self attention.(mechanism relating different positions of a single sequence). A transformer is a deep learning model that adopts the mechanism of selfattention, differentially weighting the significance of each part of the input data. It is used primarily in the fields of natural language processing (NLP) and computer vision (CV). CNNs use convolution, a “local” operation bounded to a small neighborhood of an image. Visual Transformers use self-attention, a “global” operation, since it draws information from the whole image. This allows the ViT to capture distant semantic relevances in an image effectively.

4.2.1. ADVANTAGES OF PROPOSED SYSTEM

- They hold the potential to understand the relationship between sequential elements that are far from each other.
- They are way more accurate.
- They pay equal attention to all the elements in the sequence.
- Transformers can process and train more data in lesser time.
- Transformers leverage the concept of self-attention to develop simpler models.
- A transformer is a machine learning model based solely on the concept of attention. It is a game changer in the world of deep learning architectures because it eliminates the need for recurrent connections and convolutions.

4.3. FEASIBILITY STUDY

Text summarization and image captioning are highly feasible tasks in deep learning. In fact, many state-of-the-art models for both tasks are based on deep learning architectures. For text summarization, deep learning models such as sequence-to-sequence (Seq2Seq) models with attention mechanisms have been successfully applied. These models are capable of producing high-quality summaries by learning to capture the most important information from the input text and generating a summary that conveys that information effectively. Similarly, deep learning models such as convolutional neural networks (CNNs) and recurrent neural networks (RNNs) have been used for image captioning.

These models learn to extract relevant features from the input image and generate a natural language description that accurately describes the content of the image. In recent years, there has been a significant progress in developing transformer-based models like BERT, GPT, and T5 that can

be used for both text summarization and image captioning tasks. These models have achieved state-of- the-art performance on benchmark datasets for both tasks, demonstrating the feasibility of deep learning approaches for these tasks. Economic feasibility is an important consideration when it comes to implementing deep learning models for text summarization and image captioning. While deep learning models can be computationally expensive to train, there are several factors that make them economically feasible for these tasks. Firstly, deep learning models can produce highquality summaries and captions with a high degree of accuracy, which can lead to significant improvements in productivity and efficiency. This can be particularly beneficial in industries where large volumes of text or image data need to be processed and analyzed quickly, such as news media, e-commerce, and healthcare.

CHAPTER 5

SYSTEM DESIGN

5.1. SYSTEM ARCHITECTURE

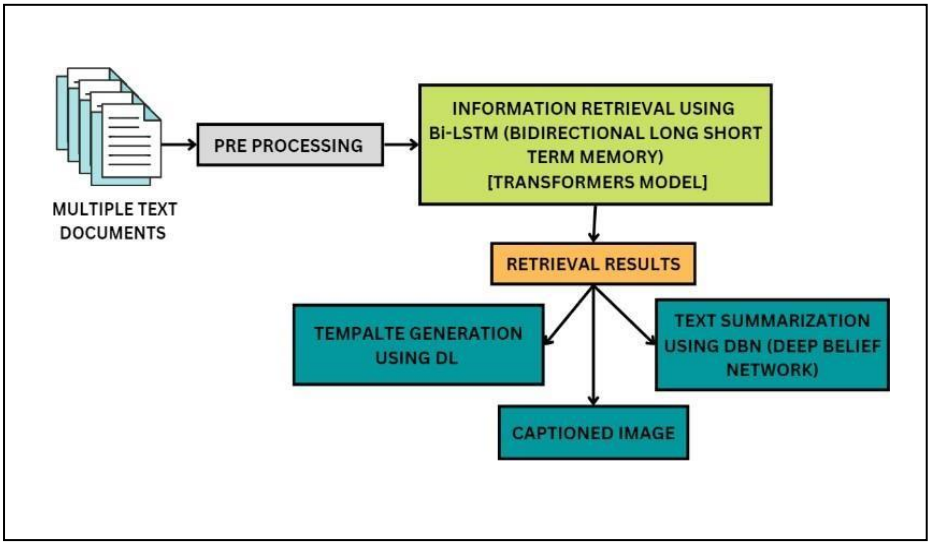

FIGURE 5.1. SYSTEM ARCHITECTURE FOR TEXT SUMMARIZATION

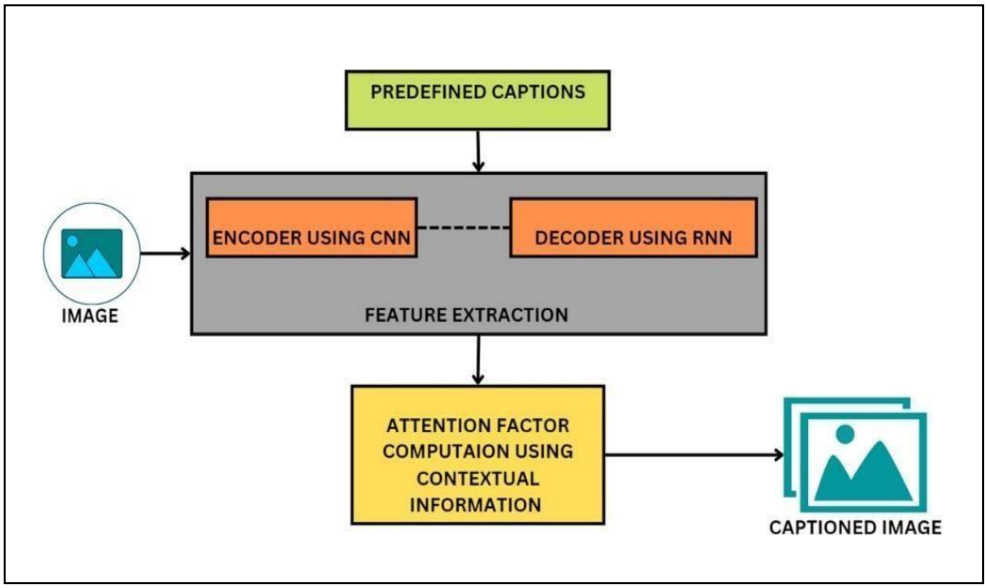

FIGURE 5.2. SYSTEM ARCHITECTURE FOR IMAGE CAPTIONING

5.2. UML Diagram

5.2.1. USE CASE DIAGRAM

A use case is a written description of how users will perform tasks on your website. It outlines, from a user's point of view, a system's behavior as it responds to a request. Each use case is represented as a sequence of simple steps, beginning with a user's goal and ending when that goal is fulfilled. In UML, use- case diagrams model the behaviour of a system and help to capture the requirements of the system. Use-case diagrams describe the high-level functions and scope of a system. These diagrams also identify the interactions between the system and its actors.

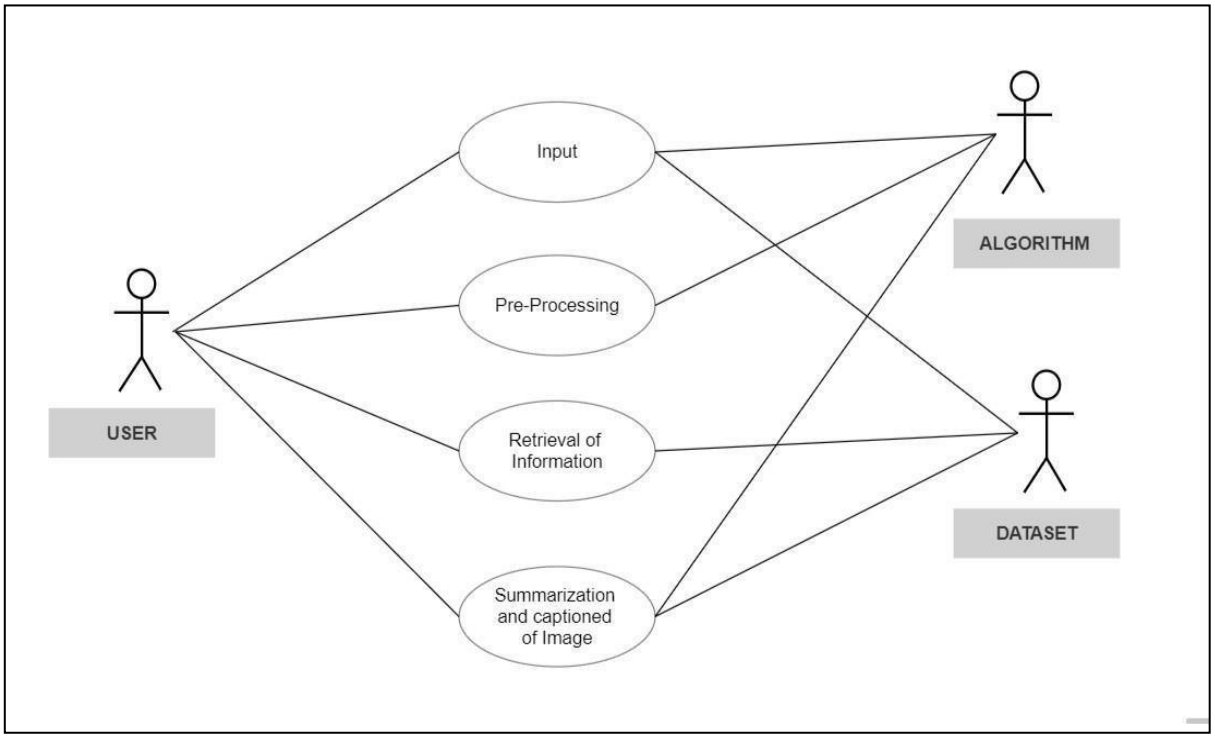

FIGURE 5.3.: USE CASE DIAGRAM

5.2.2. CLASS DIAGRAM

Class diagram describes the attributes and operations of a class and also the constraints 5imposed on the system. The class diagrams are widely used in the modelling of object-oriented systems because they are the only UML diagrams, which can be mapped directly with object-oriented languages. A class diagram is a visual representation of class objects in a

model system, categorized by class types. Each class type is represented as a rectangle with three compartments for the class name, attributes, and

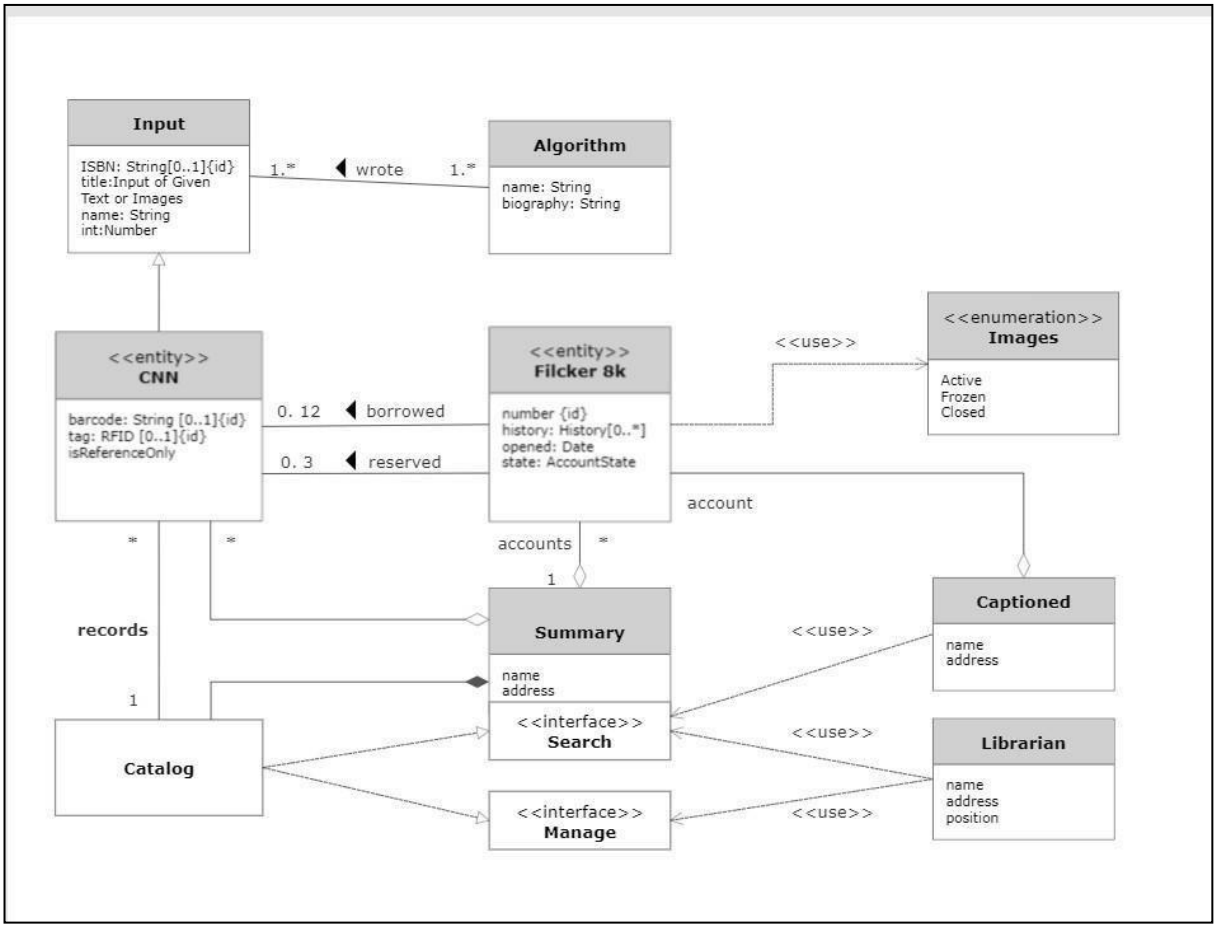

FIGURE 5.4 : CLASS DIAGRAM

5.2.3. ACTIVITY DIAGRAM

An activity diagram visually presents a series of actions or flow of control in a system similar to a flowchart or a data flow diagram. Activity diagrams are often used in business process modelling. They can also describe the steps in a use case diagram. Activities modeled can be sequential and concurrent. In UML, the activity diagram is used to demonstrate the flow of control within the system rather than the implementation.

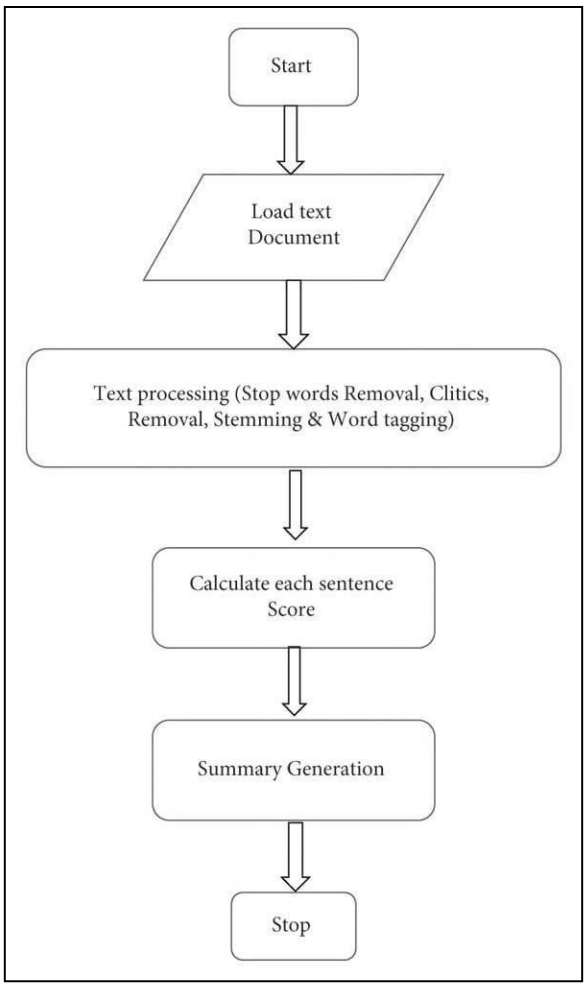

FIGURE 5.5: ACTIVITY DIAGRAM

5.2.4. ENTITY RELATION DIAGRAM

An entity relationship diagram (ERD), also known as an entity relationship model, is a graphical representation that depicts relationships among people, objects, places, concepts or events within an information technology (IT) system. An entity relationship diagram describes how entities relate to each other. In simple terms, it's a picture or a framework of your business or a certain business process. (Learn more about business process modelling). Entities are the things we need to store data about. ERDs help everyone to understand the foundations of the data/information that is going to be stored within their database.

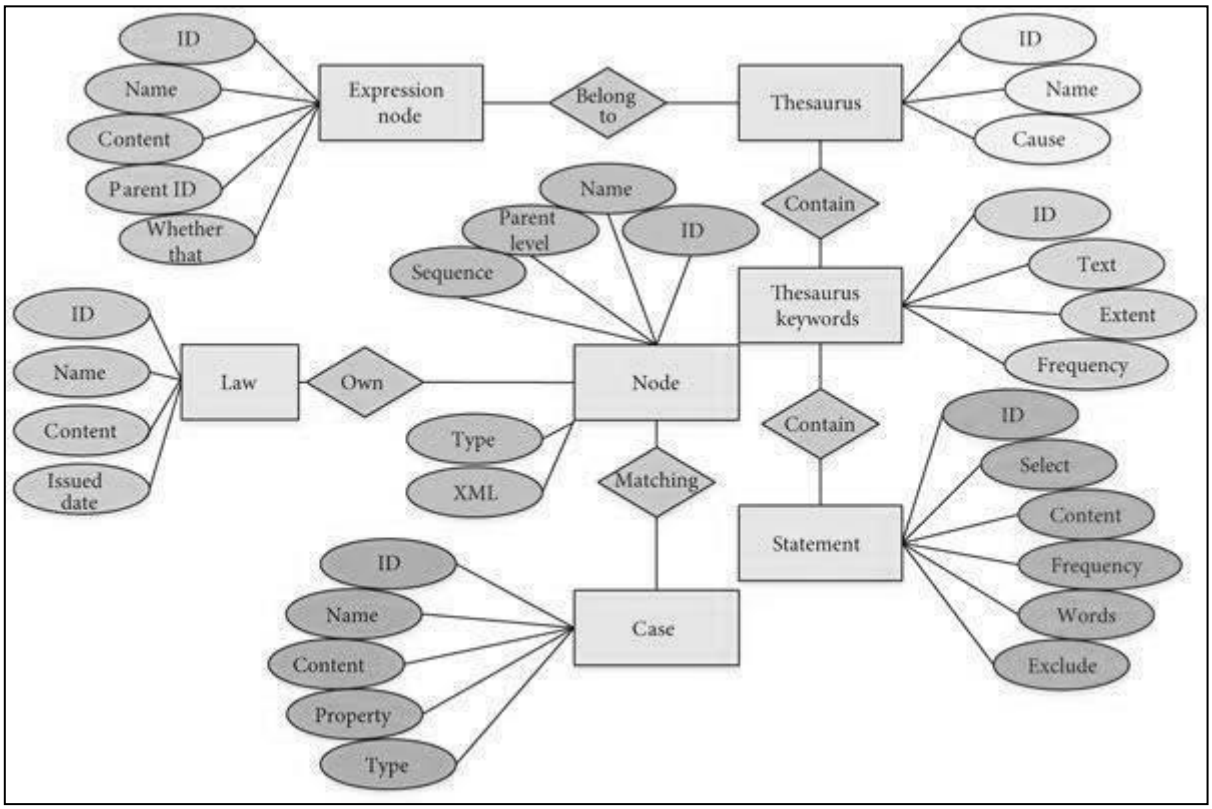

FIGURE 5.6 : ENTITY RELATION DIAGRAM

CHAPTER 6

SYSTEM IMPLEMENTATION

- Pre-processing.
- Feature Extraction .
- Image Caption Generation.
- Summarization.

6.1. DATA PRE-PROCESSING

Data pre-processing plays an important role in text summarization and image captioning tasks. Here's how it can be used in each task:

TEXT SUMMARIZATION:

In text summarization, data pre-processing involves the following steps:

Text Cleaning: The text data is cleaned by removing any unnecessary characters, such as punctuation marks, special characters, and stop words.

Tokenization: The text data is split into individual words or tokens.

Stemming/Lemmatization: Words are converted into their base forms to reduce the overall vocabulary size.

Feature Extraction: Important features are extracted from the text, such as word frequency, word co-occurrence, and word embeddings.

Text Encoding: The text data is converted into a numerical format using techniques such as bag-of-words or TF-IDF.

Text Normalization: The text data is normalized by applying techniques such as lowercasing, spell checking, and removing redundant words.

IMAGE CAPTIONING:

In image captioning, data pre-processing involves the following steps:

Image Pre-processing: The images are pre-processed by resizing, cropping, and normalizing the pixel values.

Feature Extraction: The image features are extracted using techniques such as Convolutional Neural Networks (CNNs) and transferred to a language model.

Text Pre-processing: The text data is pre-processed using techniques similar to text summarization, such as tokenization, stemming/lemmatization, feature extraction, and text normalization.

Text Encoding: The text data is encoded using techniques such as word embeddings and recurrent neural networks (RNNs) to generate captions.

Overall, data pre-processing plays a critical role in text summarization and image captioning, as it ensures that the data is in the right format and free of errors, inconsistencies, or missing values.

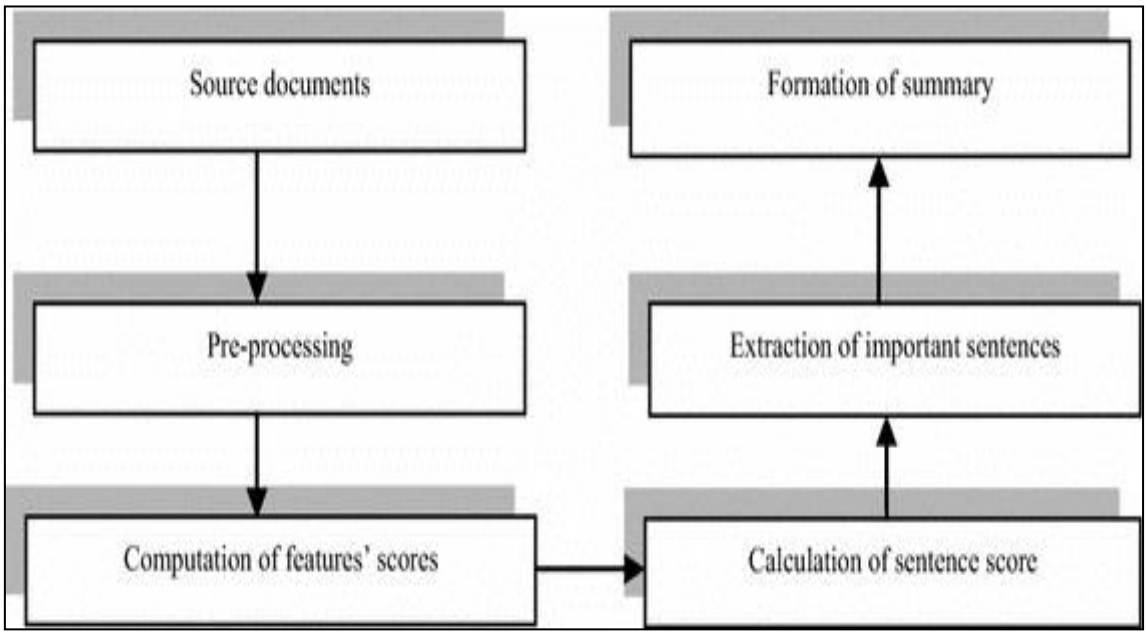

FIGURE 6.1: MODULE DIAGRAM

6.2. DATA VALIDATION/CLEANING/PROCESS

Data validation, cleaning, and processing are important steps in preparing data for analysis. These steps ensure that the data is accurate, consistent, and complete, and that it is in a format that can be easily analysed by software tools.

Data Validation:

Data validation is the process of ensuring that the data is accurate, complete, and consistent. This involves checking for missing data, outliers, and errors, as well as ensuring that the data conforms to predefined rules and constraints. Some common techniques for data validation include manual inspection, statistical analysis, and machine learning algorithms.

Data Cleaning:

Data cleaning is the process of identifying and correcting errors, inconsistencies, and inaccuracies in the data. This involves techniques such as removing duplicates, filling in missing values, correcting spelling errors, and converting data to a standardized format.

Data Processing:

Data processing is the process of transforming raw data into a format that can be easily analysed by software tools. This involves techniques such as data normalization, feature engineering, and data reduction. Data normalization involves scaling the data to a standard range or distribution, while feature engineering involves selecting and extracting important features from the data. Data reduction involves reducing the amount of data by selecting a subset of features or samples.

6.2.1. DATA ANALYSIS OF VISUALIZATION

Data analysis involves the process of exploring and summarizing data to extract insights and make informed decisions. This involves techniques such as data wrangling, data modelling, and statistical analysis. Data wrangling involves cleaning, transforming, and organizing data in a format that can be analysed. Data modelling involves creating mathematical or statistical models to predict future outcomes or understand relationships between variables. Statistical analysis involves using techniques such as hypothesis testing

6.2.2. DATA ANALYSIS OF VISUALIZATION

Data analysis involves the process of exploring and summarizing data to extract insights and make informed decisions. This involves techniques such as data wrangling, data modelling, and statistical analysis.Data wrangling involves cleaning, transforming, and organizing data in a format that can be analyzed. Data modelling involves creating mathematical or statistical models to predict future outcomes or understand relationships between variables. Statistical analysis involves using techniques such as hypothesis testing and regression analysis to analyze and interpret data. Data visualization involves creating visual representations of data to help communicate insights and patterns. This involves techniques such as charts, graphs, and interactive dashboards. Data visualization can help to simplify complex data and make it easier to understand and interpret. It can also help to identify trends and patterns that may not be immediately obvious from raw data. Data analysis and visualization are often used together to explore and communicate insights from data. For example, data analysis may identify trends or patterns that can be further explored and communicated through data visualization.

Similarly, data visualization may highlight areas of interest or concern that can be further analyzed through data analysis. Overall, data analysis and visualization are important tools for data scientists and analysts. They help to extract insights and communicate findings in a way that is easy to understand and act upon.

6.2.2. GIVEN INPUT EXPECTED OUTPUT TEXT SUMMARIZATION:

INPUT: 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).

OUTPUT: Text summarization is a common problem in machine learning and NLP.

6.3. COMPARING ALGORITHM WITH PREDICTION FORM OF BEST ACCURACY RESULT

There are several algorithms that can be used for text summarization and image captioning tasks. Comparing them based on their prediction form of best accuracy result is a useful way to determine which algorithm may be the most suitable for a particular task. Here are some examples:

Text Summarization:

Extractive Summarization: This algorithm selects the most important sentences from the original text to create a summary

Abstractive Summarization: This algorithm generates a summary by understanding the meaning of the text and creating new sentences that capture the essence of the original text.

Image Captioning:

Encoder-Decoder with Attention Mechanism: This algorithm uses a Convolutional Neural Network (CNN) to encode the image and a Recurrent Neural Network (RNN) to decode the image and generate a caption.

Show and Tell: This algorithm also uses a CNN to encode the image, but instead of an RNN, it uses a single-layer perceptron to generate a caption. The prediction form of best accuracy result for this algorithm is the generation of captions that are descriptive, coherent, and semantically meaningful, while also being grammatically correct.

Overall, the choice of algorithm for text summarization and image captioning will depend on the specific requirements of the task, such as the size and complexity of the dataset, the level of accuracy and fluency required, and the computational resources available.

6.4. DEPLOYMENT

Deployment of text summarization and image captioning models can be done using a variety of methods depending on the specific use case and requirements of the application. Here are some general steps that can be followed for deployment:

Choose a deployment platform: There are several options for deployment platforms such as cloud services like AWS, GCP, or Azure, or using a dedicated server or local machine. The choice of platform will depend on factors such as scalability, security, cost, and ease of

deployment. Prepare the model for deployment: The model needs to be saved in a format that can be loaded and used by the deployment platform. For example, a trained deep learning model can be saved in a format such as HDF5, ONNX or TensorFlow Saved Model. Additionally, any dependencies required for the model to run such as libraries, packages, and frameworks need to be installed on the deployment platform.

Define an API: An Application Programming Interface (API) needs to be defined that specifies the inputs required by the model, and the output format of the model. The API can be defined using a web framework such as Flask, Django, or FastAPI.

Deploy the model: Once the API is defined, the model can be deployed on the chosen deployment platform. This can involve creating and configuring servers, setting up load balancing and scaling, and setting up monitoring and logging systems.

Test and monitor the deployed model: Once the model is deployed, it is important to test it to ensure that it is working correctly, and to monitor it to ensure that it continues to work as expected over time.

DATA SET

6.5. PREPROCESSING THE DATASET

Preprocessing the Flicker8k dataset involves several steps that are necessary to prepare the data for use in a machine learning model. Here are some common steps for preprocessing this dataset:

- Download the Flicker8k dataset from the official website or source.
- Extract the zip file to a directory on your local machine. Open the text file named "Flickr8k.token.txt" to read the image captions.
- Create a dictionary that maps each image filename to a list of captions.

- Clean the captions by removing unnecessary characters, converting all text to lowercase, and removing any punctuation marks or special characters.
- Split the data into training and validation sets. For example, you might use 80% of the data for training and 20% for validation.
- Resize all images to a common size, such as 256x256 or 512x512, to ensure that they are all the same size before feeding them into the model.
- Convert the images to a format that can be read by your machine learning model, such as JPEG or PNG.Save the preprocessed data to a new directory on your local machine.

Overall, these preprocessing steps help to ensure that the Flicker8k dataset is clean, consistent, and ready for use in a machine learning model. The preprocessed Flicker8k dataset can be used for a variety of natural language processing and computer vision tasks, such as image captioning, object recognition, and visual question answering. Given an image, generate a natural language description of the image. This task can be achieved by training a neural network to predict a caption for an image.

6.5.1. ATTRIBUTE INFORMATION

Attribute information is the data that describes the characteristics or features of a text or an image that are relevant to a specific task, such as text summarization or image captioning. In both text summarization and image captioning, the attribute information is used to guide the generation of a summary or caption that accurately captures the most important information in the source text or image. By taking into account the relevant attributes, machine learning models can generate summaries or captions that are more informative, accurate, and coherent.the relevant attributes,

machine learning models can generate summaries or captions that are more informative, accurate, and coherent.

6.5.2. PROJECT GOALS

The goals of text summarization and image captioning projects can vary depending on the specific application and context. Here are some common goals for each task:

Text Summarization:

- To provide a concise and informative summary of a long or complex document that captures the most important information.
- To enable users to quickly understand the main ideas and key points of a document without having to read the entire text.

Image Captioning:

- To automatically generate captions that accurately describe the content and context of an image.
- To enable visually impaired users to access visual content through textual descriptions.

6.6. ALGORITHM EXPLANATION

"Flickr8k" is a dataset that is commonly used for image captioning tasks. **The algorithm used for captioning the images in this dataset typically involves the following steps:**

Preprocessing: The images are resized and preprocessed to extract relevant features such as color, texture, and shape. Common techniques used for image preprocessing include convolutional neural networks (CNNs) and feature extraction algorithms such as SIFT, SURF, or HOG.

Caption.

Generation: Once the relevant features are extracted, the algorithm generates a caption for the image. This involves using a machine learning model such as a recurrent neural network (RNN) or a transformer to generate a sequence of words that best describe the image.

Training the Model: The algorithm is trained on a large dataset of images with their corresponding captions to learn the relationships between the image features and the words in the captions. The model is typically trained using techniques such as backpropagation and gradient descent to optimize the model's parameters.

Evaluation: The algorithm is evaluated on a test set of images to measure its performance in generating accurate and informative captions. Common evaluation metrics used for image captioning include BLEU, ROUGE, METEOR, and CIDE.

The specific algorithm used for image captioning may vary depending on the specific task and dataset. However, the general approach typically involves a combination of image preprocessing, machine learning, and natural language processing techniques to generate captions that accurately describe the content and context of the image.

6.6.1. USED PYTHON PACKAGES

There are several Python packages that can be used for text summarization and image captioning tasks. Here are some commonly used packages:

Text Summarization:

NLTK (Natural Language Toolkit): a comprehensive library for natural language processing tasks such as tokenization, stemming, and summarization.

Gensim: A topic modeling package that includes algorithms for summarization, summarization evaluation, and other text processing tasks.

Sumy: A Python package that provides implementations of popular summarization algorithms such as Luhn, Edmundson, and TextRank.

Image Captioning:

TensorFlow: an open-source platform for machine learning that includes tools for developing and training deep learning models, including image captioning models.

PyTorch: a deep learning framework that includes modules for developing and training image captioning models.

Keras: a high-level neural networks API that can be used with TensorFlow or other backend engines for developing and training image captioning models.

Other common Python packages for both text summarization and image captioning include NumPy, pandas, and scikit-learn, which provide support for data processing, analysis, and visualization.

Additionally, there are specialized packages for specific types of text summarization, such as the BERT-based summarization package transformers, or image captioning, such as the image captioning package PyTorch-Image-Captioning.

ALGORITHM

6.7. TRANSFORMERS

Transformers is a deep learning architecture that was introduced in 2017 and has since become a popular approach for various natural language processing (NLP) tasks such as text classification, question answering, and text summarization. The Transformer architecture was introduced in a paper titled "Attention is All You Need" by Vaswani et al.

The Transformer architecture is based on the use of self-attention mechanisms, which allow the model to focus on specific parts of the input sequence when processing it. This makes the Transformer architecture more effective at handling long sequences compared to traditional recurrent neural network (RNN) approaches, which have difficulty with long-term dependencies.

The core of the Transformer architecture is the self-attention layer, which computes a weighted sum of the input sequence elements based on their relevance to each other.

The self-attention layer computes a query, key, and value for each input element, and then uses these to compute an attention score for each element. The attention scores are used to compute a weighted sum of the values, which produces a context vector that summarizes the input sequence.The Transformer architecture also includes feedforward layers and residual connections, which help to improve the model's accuracy and stability. The model is typically trained using supervised learning techniques, such as backpropagation and gradient descent, on a large corpus of text data. Transformers have also been adapted to image captioning tasks, where they use a combination of visual and language processing to generate captions for images.

6.8. CNN ALGORITHM

Convolutional Neural Networks (CNNs) are a type of deep neural network that are commonly used for image recognition and classification tasks. They are inspired by the structure of the visual cortex in the brain, which is specialized for processing visual information. The key feature of a CNN is its ability to learn local and spatial patterns in the input image through the use of convolutional layers. These layers apply a set of filters or kernels to the input image, which results in a set of feature maps that highlight different aspects of the image. Each feature map corresponds to a different filter and captures a specific local pattern or feature, such as edges or corners.

The output of the convolutional layers is then passed through a series of pooling layers, which down sample the feature maps by selecting the maximum or average value in a specific region of the feature map. This reduces the spatial size of the feature maps while preserving the most important features. The output of the convolutional and pooling layers is then passed through one or more fully connected layers, which perform classification or regression tasks based on the extracted features. The fully connected layers take the flattened output of the previous layers and apply a set of weights to produce the final output. CNNs are typically trained using backpropagation and gradient descent to optimize the weights of the filters and fully connected layers. The objective is to minimize a loss function, which measures the difference between the predicted output and the actual output. Overall, CNNs have shown remarkable success in image recognition and classification tasks and have been used in various applications such as object detection, face recognition, and self-driving cars.

6.9. RESULTS AND DISCUSSION:

Text summarization is the process of creating a shorter version of a longer text while preserving its important information. There are two main approaches to text summarization: extractive and abstractive. Extractive summarization involves selecting important sentences or phrases from the original text and stitching them together to create a summary. Abstractive summarization involves creating a new summary that may not necessarily use the exact words or phrases from the original text but still captures its essence. In recent years, deep learning approaches such as neural networks have been applied to text summarization, and they have shown promising results. One of the most successful neural network architectures for text summarization is the Transformer model, which was introduced in the paper "Attention Is All You Need" by Vaswani et al. (2017).

Image captioning is the process of generating natural language descriptions for images. It involves combining computer vision techniques to analyze the visual content of an image with natural language processing techniques to generate a human-like description.The most common approach to image captioning is to use a neural network architecture called an encoder-decoder. The encoder network processes the input image and generates a set of feature vectors that capture the visual content of the image. The decoder network then takes these feature vectors and generates a sequence of words that form a coherent and descriptive caption.Recent research in image captioning has focused on improving the quality and diversity of the generated captions. One approach has been to use attention mechanisms, which allow the decoder network to focus on different parts of the input image at different stages of caption generation. for longer texts and producing diverse and creative image captions.

CHAPTER 7

APPENDIX 1

A.1. SAMPLE CODE

TEXT SUMMARIZATION:

-*- coding: utf-8 -*-

TEXT.ipynb

Automatically generated by Colaboratory.

Original file is located at https://colab.research.google.com/drive/17onRqntRZvtsrE5B1sNLikCe2Aeb7uk **INSTALL MODULES**

!pip install transformers

!pip install sentencepiece

!pip install torch

IMPORT MODULES

import torch from transformers import T5Tokenizer, T5ForConditionalGeneration, T5Config model = T5ForConditionalGeneration.from_pretrained('t5- small')tokenizer = T5Tokenizer.from_pretrained('t5-small') device = torch.device('cpu') text = """Energy is one of the major inputs for the economic development of any country. All important activities concerned with present development are depending on energy in one form or other. The power generation from wind energy is technically achievable and economically feasible and its main advantages are zero fuel cost, pollution free and environment friendly. In order to increase the life of a wind turbine, it is important to estimate the reliability level for all components in the wind turbine (WT).

Unfortunately, these WTs have to contend with large failures due to the presence of considering different states with respect to the probability of failure, failure rate and the repair rate. The availability for the WTs varies from 94.45%to99% for three year (26304 hours) intervals during the years 1995-2010.This analysis yields some surprising results about some subassemblies, such as the rotor system and gear system are the most unreliable due to very high uncertainty in thewind."""

Pre-processed_text = text.strip().replace('\n','') t5_input_text = 'summarize:' + preprocessed_text t5_input_text len(t5_input_text.split()) tokenized_text=tokenizer.encode(t5_input_text, return_tensors='pt',max_length=512).to(device)

"""**SUMMARIZE**""" summary_ids = model.generate(tokenized_text, min_length=30, max_length=120)summary = tokenizer.decode(summary_ids[0], skip_special_tokens=True) summary import os import re import numpy as np

IMAGE CAPTIONING:

import matplotlib.pyplot as plt import tensorflow as tf from tensorflow import keras from tensorflow.keras import layers from tensorflow.keras.applications import efficientnet from tensorflow.keras.layers import TextVectorizationseed = 111 np.random.seed(seed) tf.random.set_seed(see d)import tensorflow as tf from keras_preprocessing.sequence import pad_sequencesfrom keras.preprocessing.text import Tokenizer from

```
keras.models import Model from keras.layers import
Flatten, Dense, LSTM, Dropout, Embedding, Activation
from keras.layers import concatenate, BatchNormalization,
Input from tensorflow.keras.layers import concatenatefrom
keras.utils import to_categorical
from keras.applications.inception_v3 import
InceptionV3, preprocess_inputfrom keras.utils import plot_model
import matplotlib.pyplot as pltimport cv2 import string import time
print("Running..... ") pip install keras_preprocessing
!wget -q
https://github.com/jbrownlee/Datasets/releases/download/Flickr8k/Flickr
8k_Dataset.zip
!wget -q
https://github.com/jbrownlee/Datasets/releases/download/Flickr8k/Flickr
8k_text.zip
!unzip -qq Flickr8k_Dataset.zip
!unzip -qq Flickr8k_text.zip
!rm Flickr8k_Dataset.zip Flickr8k_text.zip
# Path to the images
IMAGES_PATH = "Flicker8k_Dataset" # Desired image dimensions
IMAGE_SIZE = (299, 299)
# Vocabulary size
VOCAB_SIZE = 10000
# Fixed length allowed for any sequence
```

```
SEQ_LENGTH = 25

# Dimension for the image embeddings and token embeddings

EMBED_DIM = 512

# Per-layer units in the feed-forward network

FF_DIM = 512

# Other training parameters

BATCH_SIZE = 64

EPOCHS = 30

AUTOTUNE = tf.data.AUTOTUNE

def load_captions_data(filename): with open(filename) as caption_file:

caption_data = caption_file.readlines() caption_mapping = {}

text_data = []  images_to_skip = set()

[

layers.RandomFlip("horizontal"

),layers.RandomRotation(0.2), layers.RandomContrast(0.3),

])

def decode_and_resize(img_path): img = tf.io.read_file(img_path) img =

tf.image.decode_jpeg(img, channels=3)img = tf.image.resize(img,

IMAGE_SIZE)

img = tf.image.convert_image_dtype(img, tf.float32)return img def

process_input(img_path, captions):
```

return decode_and_resize(img_path), vectorization(captions) def make_dataset(images, captions):

dataset = tf.data.Dataset.from_tensor_slices((images, captions))dataset = dataset.shuffle(BATCH_SIZE * 8) dataset = dataset.map(process_input, num_parallel_calls=AUTOTUNE)

dataset = dataset.batch(BATCH_SIZE).prefetch(AUTOTUNE)

return dataset

Pass the list of images and the list of corresponding captions

train_dataset=make_dataset(list(train_data.keys()),

list(train_data.values())) valid_dataset = make_dataset(list(valid_data.keys()), list(valid_data.values())) def get_cnn_model():

base_model = efficientnet.EfficientNetB0(

input_shape=(*IMAGE_SIZE, 3), include_top=False, weights="imagenet",

)

self.num_heads = num_heads

self.attention_1 = layers.MultiHeadAttention(num_heads=num_heads, key_dim=embed_dim, dropout=0.0

)

self.layernorm_1 = layers.LayerNormalization() self.layernorm_2 = layers.LayerNormalization() self.dense_1 = layers.Dense(embed_dim, activation="relu")

```
def call(self, inputs, training, mask=None):inputs =
self.layernorm_1(inputs) inputs = self.dense_1(inputs) attention_output_1
= self.attention_1(query=inputs, value=inputs,
key=inputs, attention_mask=None, training=training,
)
out_1 = self.layernorm_2(inputs + attention_output_1)return out_1 class
PositionalEmbedding(layers.Layer):
def init (self, sequence_length, vocab_size, embed_dim, **kwargs):
super(). init (**kwargs) self.token_embeddings = layers.Embedding(
input_dim=vocab_size,
output_dim=embed_dim
)
self.position_embeddings=layers.Embedding(
input_dim=sequence_length, output_dim=embed_dim
)
self.sequence_length = sequence_lengthself.vocab_size = vocab_size
self.embed_dim = embed_dim
self.embed_scale = tf.math.sqrt(tf.cast(embed_dim, tf.float32)) def
call(self, inputs): # 7. Update the trackers
batch_acc/=float(self.num_captions_per_image)
self.loss_tracker.update_state(batch_loss)
self.acc_tracker.update_state(batch_acc) # 8. Return the loss and accuracy
values return {"loss": self.loss_tracker.result(), "acc":
self.acc_tracker.result()}
```

```
def test_step(self, batch_data):
batch_img, batch_seq = batch_databatch_loss = 0 batch_acc
= 0
 # 1. Get image embeddings img_embed =
self.cnn_model(batch_img)
# 2. Pass each of the five captions one by one to the decoder
# along with the encoder outputs and compute the loss as well as accuracy#
for each caption.
for i in range(self.num_captions_per_image):
loss,        acc    =        self._compute_caption_loss_and_acc(
      img_embed, batch_seq[:, i, :], training=False
)
# 3. Update batch loss and batch accuracy
batch_loss += lossbatch_acc+= acc batch_acc /=
float(self.num_captions_per_image) #4.
      Update        the     trackers
      self.loss_tracker.update_state(batch_loss)
self.acc_tracker.update_state(batch_acc)
#5.  Return  the  loss  and  accuracy  values  return  {"loss":
self.loss_tracker.result(), "acc": self.acc_tracker.result()}
decoded_caption  =  decoded_caption.replace("<start>  ",  "")
decoded_caption = decoded_caption.replace(" <end>", "").strip()
```

print("Predicted Caption: ", decoded_caption) # Check predictions for a few samples generate_caption()

A1.2. SCREENSHOTS

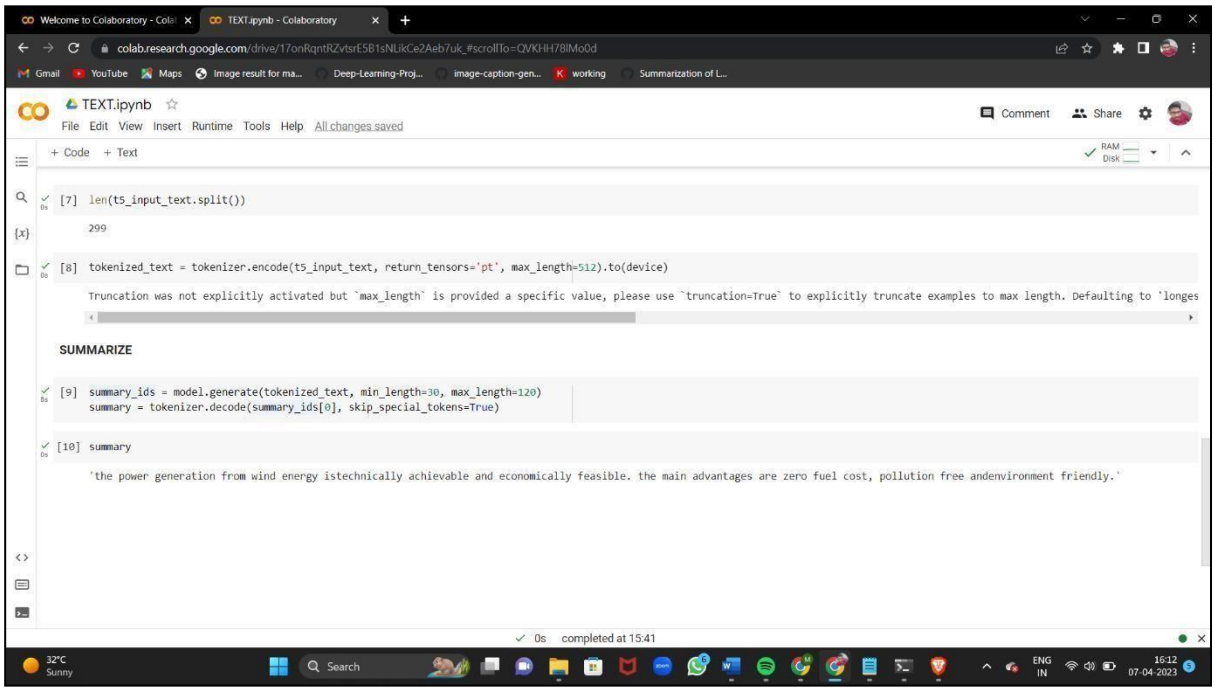

FIGURE A1.2.1 OUTPUT RESULT OF TEXT SUMMARIZATION

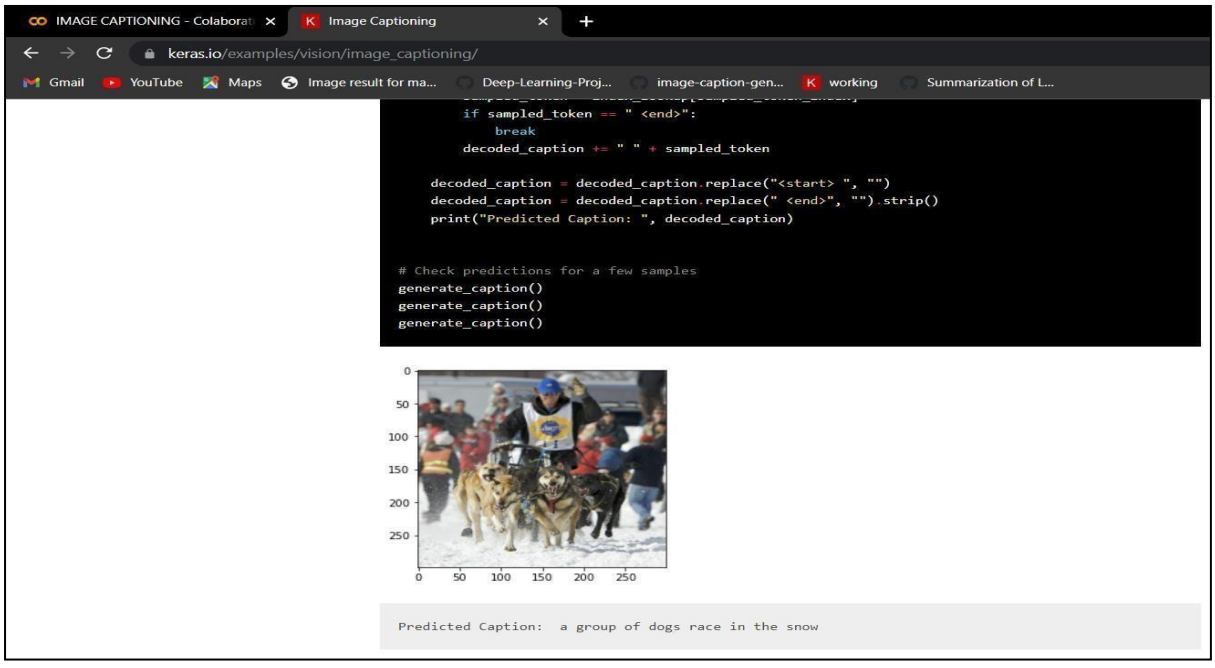

FIGURE A1.2.2. OUTPUT RESULT OF IMAGE CAPTIONING

CHAPTER 8

CONCLUSION AND FUTUREWORK

8.1. CONCLUSION

Deep learning has been effectively applied to both abstractive text summarization and image captioning. Abstractive text summarization involves using deep learning models to generate a concise and readable summary of a longer text. There are several approaches, datasets, evaluation measures, and challenges associated with this task. For example, recurrent neural networks with an attention mechanism and long short-term memory (LSTM) are commonly used techniques for abstractive text summarization. The Gigaword dataset is commonly employed for single-sentence summary approaches, while the Cable News Network (CNN)/Daily Mail dataset is commonly employed for multisentence summary approaches. Recall-Oriented Understudy for Gisting Evaluation 1 (ROUGE1), ROUGE2, and ROUGE-L are determined to be the most commonly applied metrics for evaluating the quality of summarization.

For image captioning, deep learning-based techniques are capable of handling the complexities and challenges of generating descriptions for visualized entities that exist in images. These techniques have shown promising results and have the potential to further improve the field of image captioning.

In conclusion, deep learning has been effectively applied to both abstractive text summarization and image captioning. These techniques have shown promising results and have the potential to further improve these fields. However, there are still challenges that need to be addressed in order to continue advancing these technologies.

8.2 FUTURE WORK

There are several future research trends and open challenges in the field of deep learning-based abstractive text summarization and image captioning. For abstractive text summarization, some of the challenges include the unavailability of a golden token at testing time, out-ofvocabulary (OOV) words, summary sentence repetition, inaccurate sentences, and fake facts. Future research may focus on addressing these challenges and improving the performance of abstractive text summarization models. For image captioning, future research may focus on improving the ability of deep learning-based techniques to recognize important objects, their attributes, and their relationships in an image, as well as generating syntactically and semantically correct sentences. Overall, there is still much work to be done in the field of deep learningbased abstractive text summarization and image captioning. Researchers are actively working on addressing the challenges and improving the performance of these techniques.

CHAPTER 9
REFERENCES

[1] Parmar, Chandu, Ranjan Chaubey, and Kirtan Bhatt. "Abstractive TextSummarization Using Artificial Intelligence." Available at SSRN 3370795 (2019).

[2] Gupta, Vanyaa, Neha Bansal, and Arun Sharma. "Text summarization for big data: A comprehensive survey." In International Conference on Innovative Computing and Communications, pp. 503-516. Springer, Singapore, 2019.

[3] Applications of automatic summarization : https://blog.frase.io/20-applications-of-automaticsummarization-in- the-enterprise/

[4] Shanmuga sundaram Hariharan. "Studies on intrinsic summary evaluation", International Journal of ArtificialIntelligenceand Soft Computing, 2010

[5] Kim, Joo-Chang, and Kyungyong Chung. "Associative feature information extraction using text mining from health big data." Wireless Personal Communications 105, no. 2 (2019):691-707.

[6] Bhavadharani, M., M. P. Ramkumar, and Selvan GSR Emil. "Performance Analysis of Ranking Models in Information Retrieval." In 2019 3rd International Conference on Trends in Electronics and Informatics (ICOEI),pp. 1207-1211. IEEE,2019.

[7] Pan, Suhan, Zhiqiang Li, and Juan Dai. "An improved TextRank keywords extraction algorithm." In Proceedings of the ACM Turing Celebration Conference-China, pp. 1-7.2019.

[8] Mihalcea, Rada. "Graph-based ranking algorithms for sentence extraction, applied to text summarization."