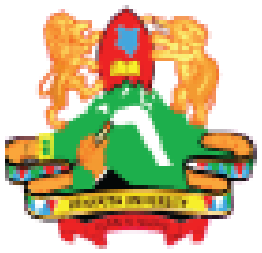


KENYATTA UNIVERSITY

DEPARTMENT OF COMPUTING & INFORMATION TECHNOLOGY

A HYBRID MODEL FOR TEXT SUMMARIZATION USING NATURAL LANGUAGE PROCESSING

JAMES MUGI KARANJA

REG: J57/37837/2016

This research proposal is submitted for the partial fulfillment of the requirements for the award of the degree of Masters of Science in Computer Science in the School of Engineering and Technology of Kenyatta University

JUNE, 2022

DECLARATION

I declare that this proposal is my original work and has not been presented in any other university/institution for consideration of any certification. This research proposal has been complemented by referenced sources duly acknowledged.

Signature: ..... Date: .....

James Mugi Karanja

Computing and Information Technology

Supervisor's declaration: This proposal has been submitted for appraisal with my approval as University Supervisor.

Signature: ……..... Date: …….....

Dr. Abraham Matheka

Computing and Information Technology (CIT)

Kenyatta University

ACKNOWLEGEMENT

This study could not have been possible without the immense support and guidance from a number of people. I wish to acknowledge their unreserved assistance towards this study.

I thank the Almighty God for sustaining me throughout this endeavor, giving me the strength, good health, courage, resources and wisdom. I also thank my family for providing me with the emotional and the financial support.

My special thanks to Dr. Abraham Matheka my supervisor for his guidance and for not giving up on me and for his unreserved support and encouragements. I would also like to acknowledge the project co-coordinator Dr. Elizaphan Maina and CIT Department Chairperson Dr. Stephen Waithaka who have assisted me throughout this piece of work.

I thank the Departments of Computing & Information Technology staff who assisted me as a student at Kenyatta University

Finally, I'm indebted to my friends especially Faith Kivuva, Anthony Wambu, Cholo Eugine and Isaac Kuria for their support in the task of reviewing, editing and illustrating above and beyond the call of duty and friendship.

TABLE OF CONTENTS

LIST OF TABLES

ABBREVIATIONS AND ACRONYMS

AI - Artificial Intelligence

NLP - Natural Language Processing

AS- Automatic Summarization

ML - Machine Learning

TF - Term Frequency

IDF - Inverse Document Frequency

GPT- Generative Pre-trained Transformer

SDTS - Single Document Text Summarization

MDTS - Multi-Document Text Summarization

QBTS - Query-Based Text Summarization

GTS - Generic Text Summarization

DSTS - Domain-Specific Text Summarization

LSTM - Long Short Term Memory

TFRS - Time Frequency Representation Summary

BART – Bidirectional and Auto Regressive Transformers

T5 Transformer-Text-to-Text Transfer Transformer

PEGASUS-Pre-training with Extracted Gap-sentences for Abstractive Summarization Sequence-to-sequence models

ROUGE - Recall-Oriented Understudy for Gisting Evaluation

RNN- Recurrent Neural Network

CNN - Convolutional Neural Networks

URL - Uniform Resource Locators

SVM - Support-Vector-Machine

DEFINITION OF TERMS

Artificial Intelligence: This is a technique in which machines/computers are made to think, behave and act like human beings.

Extractive summary: To create a summary, key sentences from the supplied content are chosen. It usually uses the word frequency technique to create the summary.

Abstract summary: This generates its own phrases and sentences in order to provide a more coherent summary, similar to what a human would provide.

T5 Transformer: This refers to text to text transfer transformer. It is an encoder-decoder model and converts all NLP problems into a text-to-text format. They make use of Transfer machine learning where pre-trained models are used to perform various different tasks thus providing higher performance.

ABSTRACT

The need for information all over the world to solve specific problems keeps on increasing daily. This pose a greater challenge as data stored on the internet has gradually increased exponentially over time. The structure of how processes used to be executed in various industries has also changed due to onset of COVID-19 epidemic. In the education sector it has led to adoption of E-learning in most institutions of higher learning. The flow to physical libraries has reduced as students are encouraged to get access to the learning materials online. This calls for a solution that can transform the huge amount of data on the internet into summarized information which can easily be understood and consumed by the recipient. This will entail the use of text summarization technique which helps in reducing a larger document into main ideas. The technique works by keeping important information while producing a summarized form of the text. Two major categories of text summarization methods exist namely: extractive and abstractive. Extractive technique concentrates on determining key themes using frequency analysis of sentence in the corpus of the text. Abstractive methods writes new summary with newly generated texts which do not appear in the corpus itself. The study will focus on both extractive and abstractive techniques of text summarizations using Natural Language Processing (NLP) and apply the use of T5 Transformer to come up with hybrid model for text summarization.

CHAPTER 1

INTRODUCTION AND BACKGROUND

1.0 Introduction

This chapter covers the background information, the problem statement, the research objectives, research questions, justification and the scope of the study.

1.1 Background Information

The volume of data online continues to grow exponentially thus pausing a great challenge on how to extract relevant information from the massive amount of data.(Nguyen et al., 2019).The technique of text summarization has been very effective in information summarization and retrieval thus helping in time saving while searching for some critical information which is relevant. This in turn helps in quick decision making. (Hassel, 2007).Text summarization is the process of using software to reduce the length of a text document so as to make a summary having important considerations from the original document. This is done by highlighting the most vital and important parts of the text.(Goyal et al., 2018) The input type determines the type of summarizer to be used. We can have a single document summarizer where the input is a small amount of text content or multi-document summarizer where the input can be derived from various sources and long documents.(Goyal et al., 2018) The complexity of the model to be created increases as the amount of text to be summarized rises.

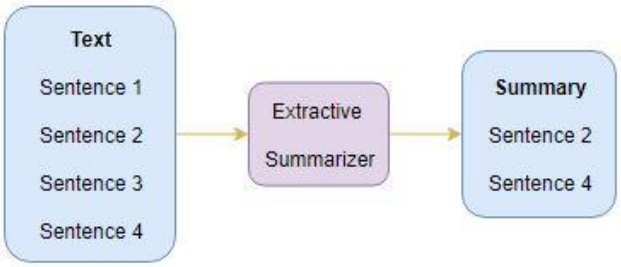
Generic summarizers treat the input without bias or prior knowledge; Domain-specific summarizers employ domain information to generate a more accurate summary based on known facts; and Query-based summarizers only contain known responses to natural language questions

regarding the input text. In Query-Based Text Summarization (QBTS), query is taken as an input in this method and depending on the query, the model is able to create the text's summary via choosing phrases and sentences which are closely related to the query posed as the input. (Boorugu & Ramesh, 2020).Domain-Specific Text Summarization (DSTS) entails use of knowledge from a specific domain like engineering or medical document which is applied in the model. The accuracy is increased and thus gives precise, more meaningful and summary that is easy to understand.(Boorugu et al., 2020) Generic Text Summarization model doesn`t take into consideration of the text meaning which needs to be summarized or the domain knowledge. A generic overview of the entire text or document is created in this method.(Boorugu et al., 2020)

In regards to the output given, the summarizer can either be extractive, in which key lines from the input text are selected to build a summary, or abstractive, in which the model creates its own words and sentences to provide a more comprehensive overview or summary, similar to what a person would produce.(Yao et al., 2018).The length of the input in Single Document Text Summarization (SDTS) is usually short. As the input for the program, only one document is issued for summarization. In the early days of text summarization, SDTS method was used. (Boorugu et al., 2020).The length of the input in Multi-Document Text Summarization (MDTS) is usually longer and numerous documents are offered as input for summary creation. MDTS is usually difficult than SDTS because you have to integrate multiple documents' summaries into a single document. Sometimes, there might be difficulty because of the diversity of themes in different documents.(Boorugu et al., 2020)

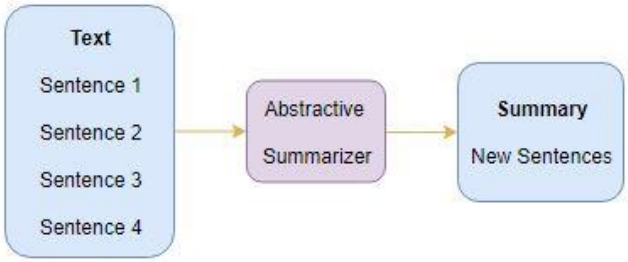
Extractive methods create a summary from the source material by extracting important phrases, keywords and sentences. By succinctly paraphrasing the source text, abstractive approaches

provide a summary that resembles a human being written abstract. Extractive method assures that the created summaries are grammatically and semantically correct, while abstractive method generates more diverse and new content. The process of generating abstract summaries is a more difficult undertaking as compared to extractive approaches. All of these methods are still a long way from being human-like but with the advancement of technology in machine learning powerful models are being developed. (Yao et al., 2018)



(Yao et al., 2018)

Figure 1 : Extractive Summary creation



(Yao et al., 2018)

Figure 2 : Abstractive Summary creation

Boorugu et al., (2020)describes Extractive Text Summarization as the process of extracting or mining sentences or phrases from a larger text to produce a condensed version with similar

meaning to the original. The majority of today's models are extractive in character. Abstractive Text Summarizing is an advanced method of summary creation that produces summarized content that isn`t in the main text but has the same meaning as the entire text. It is more challenging to develop models that form phrases or sentences that convey the similar meaning as the original text using this strategy.(Boorugu et al., 2020)

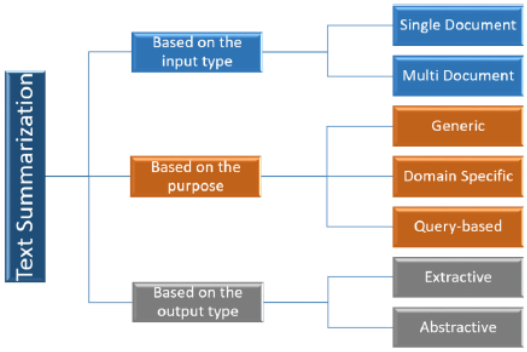


Figure 3: Text Summarization

(Boorugu et al., 2020)

The online content is massive and calls for invention and improvement of models used for content summarization. Shimada et al., (2018) gives us the importance of having a model that can help in summarizing learning materials. The model is said to help in retrieval of important points from a long text easily thus reducing the time spent to extract the points. This equips students with reading and comprehension materials that take less time. Wei et al., (2021) mention the huge volume of opinions available online which needs summarizing in order to extract the relevant information. With learning in most of the institutions of higher learning been

moved to online, there is greater need to having a means of online information summarization to improve the process of information retrieval. (Wei et al., 2021)

1.2 Transformers

This refers to an encoder-decoder model which converts all NLP (Natural Language Processing) problems into text format. It aims to solve sequence-to-sequence assignments while tackling long-range dependencies with ease(Prateek, 2019). Transformers make use of Transfer machine learning where pre-trained models are used to perform various different tasks thus providing higher performance. In Transfer Learning, the model is first trained on a task with a lot of text before being fine-tuned on a downstream job, allowing it to gain general-purpose abilities and knowledge that can be used to tasks like summarization(Zhuang et al., 2021).

Lewis et al., (2019) discussed Bidirectional and Auto Regressive Transformers (BART), which are created with a seq2seq model that has been pre-trained with de-noising. It employs a conventional seq2seq model architecture, which includes a BART-like encoder (Devlin et al., 2018) and a GPT(Generative Pre-trained Transformer) like decoder. Gupta et al, (2021)highlighted about T5 Transformers in his study. This refers to text to text transfer transformer. It is an encoder-decoder model and converts all NLP problems into a text-to-text format(Gupta et al., 2021). Another type of transformer is discussed by Zhang et al., (2020). This model is called PEGASUS and stands for Pre-training with Extracted Gap-sentences for Abstractive Summarization Sequence-to-sequence models. It works by selecting the important sentences at random as the input is passed through several encoder decoder channels.(Zhang et al., 2020)

1.3 Problem Statement

The need for information all over the world to solve specific problems keeps on increasing daily. This poses a greater challenge as data stored on the internet has gradually increased exponentially over time. (Nguyen et al., 2019)The structure of how processes used to be executed in various industries has also changed due to onset of COVID-19 epidemic. In the education sector it has led to adoption of E-learning in most Institutions of Higher Learning as students are encouraged to get access to research and learning materials online instead of visiting the physical libraries. With the vast amount of data online, extracting meaningful information might not be easier .(Goyal et al., 2018) This might lead to a lot of time utilized during a research and learning as students go through various learning materials online. This call for a solution that can help transform the huge amount of data into summarized information which is easily consumable. Research previously conducted concentrated on summarization of only plain text. This calls for development of a hybrid model for text summarization that will be able to extract and summarize text from various sources i.e pdf, plain text, and website.

1.4 Objectives

1.4.1 General Objectives

The aim of this project is to develop a hybrid model for text summarization using natural language processing.

1.4.2 Specific Objectives

i. To investigate current state of text content summarization

ii. To apply T5 Transformers in developing text content summarization model.

iii. To develop hybrid model for text summarization using extractive and abstractive summarization methods.

1.5 Research Questions

i. What is the current state of text content summarization?

ii. What is the current application of Transformers in developing text content summarization systems and how effective are they?

iii. Are there existing hybrid text content summarization models which utilize both abstractive and extractive techniques and how effective are they?

1.6 Justification

The data available online is enormous. A lot of time can be consumed while trying to extract meaningful information from the huge amount of data available. This project will try to come up with a way of summarizing the text data available online so as to fast track the learning and research process and make information easily consumable.

1.7 Scope

The study will revolve around development of a hybrid model for text summarization using natural language processing. The system should be able to summarize English text content obtained from the website i.e. Wikipedia, pdf document and also from text pasted into the system.

CHAPTER 2

LITERATURE REVIEW

2.0 Introduction

This chapter analyzes related works, techniques for text summarization, various models in use, impact of using automatic text summarization systems.

2.1 Text Summarization

Summary creation process necessitates first reading and comprehending the original content. The main components of the paper are stated based on the known events, facts, or situations to satisfy the summary's objective. The summary would not include all of the material in the original text, but only the parts that were judged relevant. This is self-evident, given that the summary's objective is to decrease the quantity of information in the source papers. The summary is subsequently prepared in a suitable output format after defining the main aspects inside a document. In general, there are three components to summarization: input, analysis, and output. Humans often require a comprehension of the native language in which the text material is written in. The goal of the summary and the target audience would both need to be determined throughout the analysis. The final step before the summary is delivered to the user would be to create a proper output format for the summary. (Radev & Erkan, 2015)

There are several things to consider regardless of whether the summarizer is a person or a machine. In terms of input: The summarizer would have to select how to approach reading the material based on how it is structured. For example, information relevant in the analysis stage may be found in the headers of chapters or the labels of Figures and Tables. Some document metadata, such as keywords in HTML websites, may also be useful. If the document was classed and the summarizer has access to the class or domain to which it belongs, it may be feasible to

use domain-specific information to help in the analysis and output phases. The summarizer may be influenced by the language used in the text documents. Human summarizers normally need to know the language in which the material was written.(Yao et al., 2018)

Humans have a comprehensive and profound knowledge of natural language, something machines do not have. Furthermore, human summarizers frequently have previous knowledge and common sense about the world, as well as the document's subjects, allowing them to infer between phrases.(Wei et al., 2021)

The summary's output might take several distinct formats. The summary can take the form of excerpts from the original text which is unchanged, such as whole phrases or paragraphs. It can also take the form of abstractions, which are made up of new words or sentences. The length of the summary varies as well, depending on the intended purpose and desired compression rate. The summary might take the form of whole sentences or simple words like those found in news headlines. Depending on the user interface, summaries can be provided as plain text or with additional contextual information such as hyperlinks or related phrases. (Mihalcea, 2004)

2.2 Related Works

Every day, massive volume of information is published on the internet.(Boorugu et al., 2020) This necessitates the use of a solution that can assist in the transformation of large amounts of data into summarized information. This will necessitate the application of a text summarizing technique, which aids in the reduction of a massive material into key points. The method works by keeping important information while producing a condensed version of the text.

Automatic summarization (Chu et al., 2015) or indexing (Garg et al., 2015) has been investigated in a variety of study disciplines, including video, audio, and document (text) processing. In a number of studies, text summarizing techniques were employed to construct concise summaries

of documents. A frequency thresholding method was developed in an early text summarizing study. The term frequency, inverse document frequency (TF-IDF) method (Shimada et al., 2018), which has been found to attain a reasonable degree of performance, has been introduced to improve frequency-driven approaches. TF-IDF concentrates only on extractive text summarization where it uses word frequency to generate summary. Even though some words might have higher frequency, they might not be the main points thus leading to misleading summary.

Eberts et al., (2015)suggested a system for automatically condensing educational video content. Their method pinpoints the exact moment and location in video footage where presentation slides appear. In their method, they combine image processing and machine learning approaches. They also created electronic lectures and screencasts with the technology. The findings suggested that the summaries created gave viewers more information. (Eberts et al., 2015) The research concentrates on video content and fails to address summarizing text content which is usually used in most institutions.

A method for summarizing oral lectures was proposed by Chen et al., (2011). On a graph created, a random walk is performed making use of automatically extracted key phrases and latent semantic analysis with probabilities in their method. They used their method to obtain each document summary from lecture documents.(Chen et al., 2011) The researcher failed to elaborate more on how to extract the content from various sources in order to generate the summary. Li et al., (2014)suggested a completely automatic approach for capturing the full presentation utilizing camera techniques such as panning, tilting, and zooming to extract the semantic structure of a typical academic lecture video (Li et al., 2014) .General video summarizing approaches were shown in order to obtain more precise display structures than their

system. This concentrates on video content summary but doesn`t talk of text content summary generation which is easily accessible to large population.

It is sometimes suggested that studying ahead of time of a class is critical in order for students to familiarize themselves with key words from the study, and learn new concepts and terminology. Shimada et al., (2018) stressed the necessity of giving students a glimpse of what they will study ahead of time. Furthermore, effective preparation before lectures starts has been linked to improved understanding and grasping of the concepts taught during the lecture. Students are frequently requested to study a textbook or preview content in order to prepare for their next class at universities (Shimada et al., 2018) .There is need of having a way of creating summarized content which can be used in study preview instead of going through a whole topic in a textbook.

2.3 Surface-level Features

These are techniques which rely on surface-level characteristics retrieved from test documents. They were one of the initial techniques to be used in text summary generation area, dating back 1950s. These methods are often extractive, relying on factors such as sentence location, word frequency, the existence of words and phrases to be considered, and tittle words. (Meena et al., 2020)

2.3.1 Word Frequency

Approaches that depend on word frequencies assume that key areas of text will most likely contain terms that occur frequently. This strategy is implemented in the first summarizing approach presented by Meena et al., (2020) .His approach is based on the assumption that frequently recurring phrases indicate the document's core subject. Sentences are given ratings

based on the number of words they include. After evaluating all sentences based on their scores, the top ranking sentences are selected to construct a summary.(Meena et al., 2020)

The given words frequencies weight in Brandow et al., (2014) is influenced by their document placements. They used the terminology frequencies to define what they were talking about in Kupiec et al., (2015).Thematic Words is a term used to describe a group of words that are related to Each phrase contains a modest number of topic words. The quantity of theme words affects the final score. Topic groupings are discussed in (Hovy & Lin, 2013).

2.3.2 Position

Many summarizing techniques also make use of the position of sentences within a text. When rating sentences, the structure of the material is frequently taken into consideration. For certain approaches, the first few phrases of a text are considered more essential than the rest. This is consistent with the findings of the Baxendale, (2014) research, which looked at 200 paragraphs. The theme sentence was found to be the first sentence in 85 percent of the investigated paragraphs and the last sentence in 7 percent. Edmundson, (2016) also discovered that theme phrases appear extremely in the beginning or at the end of manuscripts.

2.3.3 Words and phrases to consider (Cue)

The inclusion of particular words in a sentence might depict whether the statement is important or not. Bonus and stigma terms were defined in the work of Edmundson, (2016). The presence of bonus words in a phrase, such as noteworthy, suggests its importance. Stigma words like "hardly" or "impossible" signal that a statement is irrelevant, making it more likely to be omitted from the summary. Pollock & Zamora, (2017) introduced the ADAM technique for eliminating sentences from summaries depending on the presence of particular terms. Annual lists of cue

phrases were created in Teufel & Moens, (2015) to signal the importance of a statement (Aone et al., 2018) .

2.3.4 Title words

Words occurring in the document`s title are identified using this approach. Sentences that share terms with the title are given more weights. (Hovy et al., 2013)

2.4 Hybrid summarization

Combining more than one feature of text summarization leads to development of hybrid model. Rani et al., (2017) proposed a hybrid model which makes use of the word frequency and the location of the paragraphs. Words contained in paragraphs at the beginning of the document are given more weights than words in the preceding paragraphs. Even though the model developed is hybrid, it only makes use of extractive technique of text content summarization.

Combining machine learning clustering techniques, a hybrid approach for extractive document summarization is provided .The system cascade the clustering technique's summary with SVM (Support-Vector-Machine) in order to increase its performance and quality(Patil et al., 2014). This system is hybrid because it combines various aspects of text summarization but the author only concentrated on extractive technique alone.

MuraliKrishna et al., (2013) offer a hybrid summarizing system in which sentences are extracted from documents based on the sentence scoring method. The average of the values evaluated using statistical and linguistic methodologies is used to calculate the sentence scoring method. The duplicate information in these retrieved sentences is handled using an iterative clustering process. The final result, known as the document summary, provides the most important sentences in the text related to the query without redundancy. The generated sentences can be sorted by their sentence score or the order they were presented in the original source. The

researcher make use of extractive method of summary generation and doesn`t mention abstractive technique.

A hierarchical hybrid multi document model using extractive technique was invented by Celikyilmaz & Hakkani-Tur, (2010) . The model is designed with the capability of splitting document into major subtopics which in turn are divided into more small topics. The content in the subtopics is summarized independently via extractive technique and the results combined together to form a final summary. The outcome of this system would be more fine-tuned if they would have been channeled to a model using abstractive technique. Even though the research developed a hybrid model, it only made use of one major technique of content summarization.

A multiple text document system which uses hybrid summarization techniques was developed by Dave & Jaswal, (2016). The system uses extractive techniques in the initial stage and the output are channeled to an abstractive model which uses Word Graph generation which locates important nodes from the extractive summary using heuristic rules. Heuristics rules aid in easier problem solving and provide a shortcut to solving difficult problems but don`t necessarily give an optimum solution (Wikipedia, 2022). This might affect the output of the summary. The system only tackle summarization of text content pasted on it and doesn't provide the allowance of summarization of online text content i.e from Wikipedia. Instead of using Word Graph, the research proposes use of T5 transformers for the abstractive summary generation so as to improve the precision of the output summary and also provide the allowance of getting the text content to be summarized from various sources i.e extracted from pdf document and scrapped from website URL(Uniform Resource Locators) like Wikipedia.

2.5 Techniques for text summarization

Several techniques for summary generation have been invented. Generic summarizers treat the input text without bias or prior knowledge; Domain-specific summarizers employ domain resources or data to generate a very precise and accurate summary depending on existing information; and Query-based summarizers only contain known responses to questions in natural language regarding the input data(Boorugu et al., 2020). The question is used as an input in Query-Based Text Summarization (QBTS), and the model is able to construct a synopsis of the material in response to the inquiry by picking phrases and sentences that are closely connected to the query. Boorugu et al., (2020) Domain-Specific Text Summarization (DSTS) refers to the application of knowledge from a specific domain, such as engineering or medical documents, to a model. The accuracy improves, resulting in a more exact, meaningful, and easy-to-understand summary. (Boorugu et al., 2020).The generic text summarization model ignores the text's meaning that has to be summarized as well as domain knowledge. This method generates a summary which is generic of the entire text or document. (Boorugu et al., 2020)

The summarizer is said to be extractive, where relevant sentences derived from the original text are selected to construct a summary or abstractive if the model develops its own words and sentences to provide a very comprehensive synopsis, similar to the one human being can develop (Yao et al., 2018).In Single Document Text Summarization (SDTS), the input is typically brief. There is only one document used as the summarization input. In the early days of text summarization, SDTS was used. (Boorugu et al., 2020)

In Multi-Document Text Summarization (MDTS), the input is frequently longer, and numerous documents are supplied as input for the development of the summary. Since aggregate summary

of multiple documents into one single document must be done, MDTS is usually more challenging than SDTS. Due to the diversity of themes in different documents, it is more challenging.(Boorugu et al., 2020)

As machine learning technology advances, more powerful models are being produced (Yao et al., 2018). Boorugu et al., (2020) defined Extractive Text Summarization as the process of extracting or mining sentences or phrases from a whole text that have same meaning to the original entire text but in a shortened version. Most of today's models are extractive in character. Abstractive Text Summarizing a more sophisticated method of summary generation that produces content that isn`t in the source document but have the similar meaning as the entire text. It is more difficult to develop models that form phrases with similar meaning as the original text using this strategy (Boorugu et al., 2020). Despite all these techniques for summary creation being put in place, they haven`t managed to make the text contents available easily consumable.

2.6 Text summarization models

2.6.1 Term Frequency (TF) - Inverse Document Frequency (IDF)

The TF-IDF algorithm was utilized in the extractive summary generation. Term Frequency (TF) is used to count the frequency of the words. The frequency discovered is used to assess the word's significance. The more frequently a term appears in a document, the more important it is. The straightforward explanation for TF is that it counts frequency in which a word emerges or seen in a document (Meena et al., 2020). IDF provides unique words a higher value and repeated words a lower value. TF occasionally overestimates the significance of stop words depending on how often they appear. Inverse Document Frequency determines the rarest of words that appear in the document to address TF's issue. IDF is the inverse of TF; when the two are combined, the

result is TF-IDF refers to the product of TF and IDF. The derivation of the formula is shown below.

tf(i, j) =term (i) within the document j

$$(i, j) - \log^{N} \bar{df}_i$$

TFIDF (i, j) = tf(i,j) * idf (i, j)

The equation above shows the frequency of the phrase *i* in the record *j* (Meena et al., 2020). The dataset's overall document count is N, and the records that include the word at the very least once are *dfi,* when a word is regularly utilized in several papers, its value increases. The word value *i* in the document *j* of the document N is TF–IDF. TF–IDF is a powerful algorithm for summary generation though it is only applicable for extractive summary generation and can`t rewrite the summary using different words other than the one in the main document.

2.6.2 Text Rank Algorithm

Text rank is one of an unsupervised method that uses weights as a value to rate sentences. The foundation of the text rank algorithm may be traced back to page rank on Google system, which performs the ranking of websites in regard to their links and their significance (Mihalcea, 2004). As the name implies, a directed graph is built using phrases. This is referred to as ranking method which is graph-based. The phrases are referred to as nodes or vertices, and edges are used to connect nodes that are related (Mallick, et al., 2019). The text rank algorithm is a referrer -based system in which the vertices joined by the edges recommend the relevance of the phrases in the graph. The weights assigned to the sentences are used to rank the sentences and summary is generated from the sentences having more weight. Text Rank Algorithm employs only

extractive method of summary creation. A system employing both extractive and abstractive summary generation models can be of greater help in gauging the validity of the summary created.

2.6.3 Sequence to Sequence Model

Time Frequency Representation Summary (TFRS) method employs an abstract summary model known as a sequence to sequence model to improvise new words while maintaining the meaning of the original text. The Sequence to Sequence approach, which currently powers apps like Google Translator, image captions assignment, text summary creators and internet chat bots, was initially introduced by Google.. It uses an encoder-decoder model which translates sequences of varying lengths as input and output (Song et al., 2019). The encoder-decoder component's Long Short Term Memory (LSTM) is helpful for identifying durable relationship. The training and inference phases of the encoder-decoder model are separated. In the training and inference phases, both the encoder and the decoder are utilized. During the training phase, the encoder reads the whole input sequence word by word, processes it, and stores the results in a hidden state. The hidden state of the encoder is used to train the decoder to anticipate the next word in the sequence based on the hidden state of the preceding word (Kostadinov, 2019).During the inference phase, the sequence to sequence model is tested on fresh sequences for which the target summary sequence is unknown(Nallapati,et al., 2016). Sequence to sequence models has provided feasible solutions for abstractive summarization but is still hard to tackle long text dependency in the summarization task. (Liao et al., 2020)

2.6.4 Transformers for Text Summarization

Transformer is an encoder-decoder model and converts all NLP problems into a text format. They make use of Transfer machine learning where pre-trained models are used to perform

various different tasks thus providing higher performance(Gupta et al., 2021). The construction of a transformer model is shown in the diagram below. The picture shows the different normalization and multi-head attention layers, as well as an encoder and decoder layers which make up a transformer model.

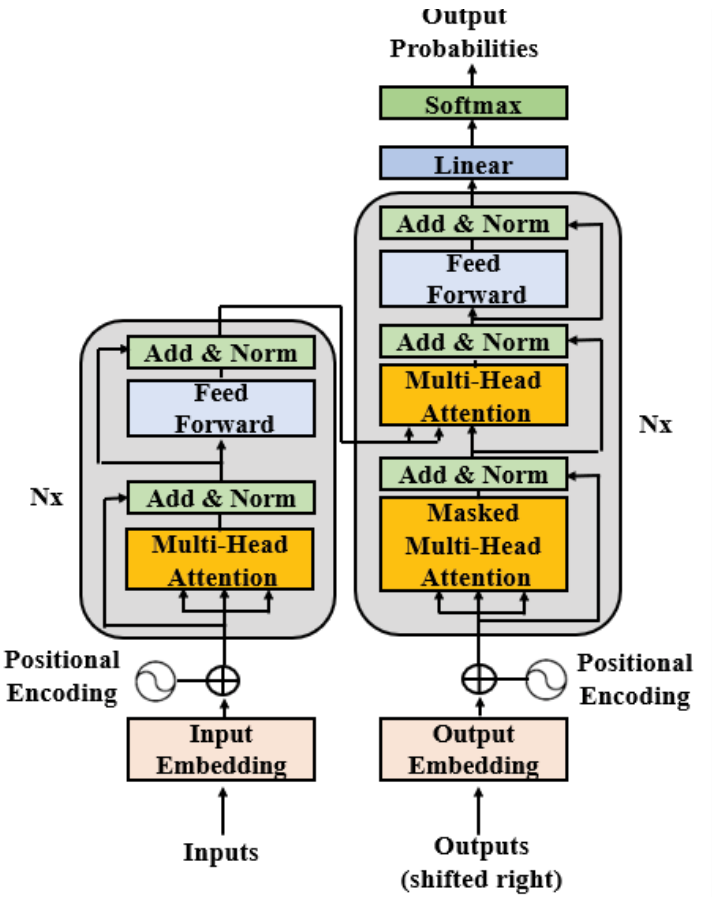


Figure 4: Transformers

(Gupta et al., 2021)

The encoder is on the left, while the decoder is on the right.The modules that make up both Encoder and Decoder can be layered on top of one another several times given as Nx. Gupta,et al., (2021) states that transformer network relies primarily on many levels of attention which

makes up majority of the modules. Since we cannot utilize long phrase directly, the inputs and outputs are first embedded into an n-dimensional space. Each word is assigned a relative position as the sequence relies on the elements order. For memorizing the word order in the input sequence, it does not employ RNN (Recurrent Neural Network) and instead relies on attention layers and positional encoding. The global dependencies formed by using several attention layers aid in the parallelization of input processing. Encoder and decoder layers are coupled to a multi-head attention layer and feed forward network levels in the transformer model. The model uses cosine and sine functions to recall the location and sequence of words, resulting in positional encoding. The encoder and decoder layers use a multi-head attention layer and applies a mechanism called self-attention (Gupta et al., 2021). The input is keyed into 3 linked layers to generate query (Q), key (K), and value (V) vectors. The vectors are subdivided into n vectors.

$$\text{Attention } (Q, K, V) = \text{softmax}\left(\frac{Q\,K^T}{\sqrt{d_k}}\right)V$$

The attention weight is determined by the effect of all other words in the sequence marked by K on each word of the sequence denoted by Q. Q, K, and V are different depending on if the attention modules are in the encoder or decoder in the structure. (Maxime, 2019)

The three major types of transformers are T5 Transformers, BART(Bidirectional and Auto Regressive Transformers) transformers and PEGASUS (Pre-training with Extracted Gap-sentences for Abstractive Summarization Sequence-to-sequence) (Gupta et al., 2021).

Transfer learning is the concept behind the Text-to-Text Transfer Transformer T5 model (Zhuang et al., 2021). In Transfer Learning, the model is first trained on a task with a lot of text

before being fine-tuned on a downstream job, allowing it to gain general-purpose abilities and knowledge that can be used to tasks like summarization. T5 employs a sequence-to-sequence creation technique, in which the encoded input is fed to the decoder through cross-attention layers, and the decoder output is autoregressive. The encoder block includes a self-attention layer and a feed forward network, according to Gupta et al., (2021). Except for a generalized attention mechanism after each self-attention layer, the decoder and encoder are structurally comparable. The model can only work with the prior outputs. The output from the last decoder block is passed into another layer.

Lewis et al., (2019) discusses Bidirectional and Auto Regressive Transformers (BART), which are created with a seq2seq model that has been pre-trained with de-noising. It employs a conventional seq2seq model architecture, which includes a BART-like encoder (Devlin et al., 2018) and a GPT(Generative Pre-trained Transformer) like decoder . The pre-training job entails a novel method in which text ranges are exchanged with only a single mask token, as well as modifying the order of the original phrases at random. BART's (Lewis et al., 2019)big model has double the number of layers as the base model. It closely resembles the BART model (Devlin et al., 2018); however BART has around 10% more features than a comparable BART model. The decoder in BART is autoregressive and is programmed to generate sequential NLP tasks like text summarization. The information is obtained given the source but altered, which is connected to the pre-training goal of de-noising. As a result, the encoder receives the input sequence embedding, and the decoder creates output auto regressively.

The construction of the decoder and encoder is identical, with the exception that after each self-attention layer, there is a generalized attention mechanism. This enables the model to just work on the previous results. The last decoder block's output is passed into another layer. Important

lines are removed from the source text and assembled as distinct outputs in Pre-training with Extracted Gap-sentences for Abstractive Summarization Sequence-to-sequence models (PEGASUS)(Zhang et al.,2020). Furthermore, selecting just relevant sentences trumps selecting sentences at random. This approach is ideal for abstractive summarizing since it is equivalent to understanding the full content and producing a summary. It's used to create the PEGASUS model by training a Transformer model on text data. The CNN/DailyMail summary datasets were used to train the algorithm. Gupta et al., (2021) did a research on several Transformer models for text summarization and found out that T5 outperformed all other models by comparing the Rouge scores for each model as shown in the table below.

Models	**Evaluation Metrics**		
	ROUGE-1	***ROUGE-2***	***ROUGE-L***
Pipeline - BART	0.38	0.28	0.38
BART modified	0.40	0.28	0.40
T5	**0.47**	**0.33**	**0.42**
PEGASUS	0.42	0.29	0.40

Figure 5: Quantitative analysis report

(Gupta et al., 2021)

ROUGE (Recall-Oriented Understudy for Gisting Evaluation) Score refers to metric that automatically help in determining the accuracy of synopsis by comparing it with summary generated by humans or comparing it with summary generated by another model. This metrics count how many overlapping units like n-gram, word sequences, and word pairs between the models created synopsis and the human generated synopsis(Tsuchiya, 2018). ROUGE-1, ROUGE-2 and ROUGE-L values are usually got in the analysis where we compare the precision and recall parameters. ROUGE-1 Precision and Recall compare the similarity of uni-grams among reference synopsis and model generated summary. Every token of comparison is a single

word. ROUGE-2 Precision and Recall compare the similarity of bi-grams among reference summary and model generated summary. Every token of comparison is two consecutive words. ROUGE-L Precision and Recall measures the Longest Common Subsequence among reference summary and model created synopsis. This refers to tokens which are in sequence but should not always be consecutive.(Tan, 2022)

Transformer models produce abstractive summary which is similar to human made summary. Even though the summary is closer to what human can generate, transformer models requires more processor power to generate the summary as compared to Extractive summary generation. This is because the T5 transformer reads through the whole document and tries to generate a summary. The longer the document, the longer it takes to generate a summary. Extractive models generate the summary very first and require less processing power even though the summary generated usually has some inconsistency. Both extractive and abstractive summary generation models have their strengths and shortcoming. When the two methods are combined, they can complement each other .If the output of extractive model is keyed in as the input of the abstractive model, more refined summary can be generated. This calls for having a hybrid model for text summarization which will put into consideration both extractive and abstractive summary generation models and also help in extracting text content from various sources i.e internet, pdf documents. The hybrid model for text summarization will also help in gauging the validity of the summary created from both models used. This will be conducted using ROUGUE (Recall-Oriented Understudy for Gisting Evaluation) Score analysis.

CHAPTER 3

RESEARCH METHODOLOGY

3.0 Introduction

This section describes the principles, rules and stipulated procedures to be used in the project. This section will focus on project design, system requirements, technology used, system functionality and system development process.

3.1 Methodology

An Iterative Incremental Methodology will be adopted in the system development. Several testing and reviews will conducted at each step thus enabling system specifications to be error free and reliable. Testing and debugging will be conducted on smaller iteration hence making the task easy and effective compared to testing complete system requirements at once. At the beginning, simpler implementations of a hybrid model for text summarization will be designed and developed. Additional functionalities will be incorporated in the system in various iterations. The design modifications will be made and new functional capabilities added in every cycle. The iterations will continue until the development is completed. This methodology enables the user to evaluate the system functionality periodically until the final product is delivered. This creates rooms for capturing new requirements and implementing them.

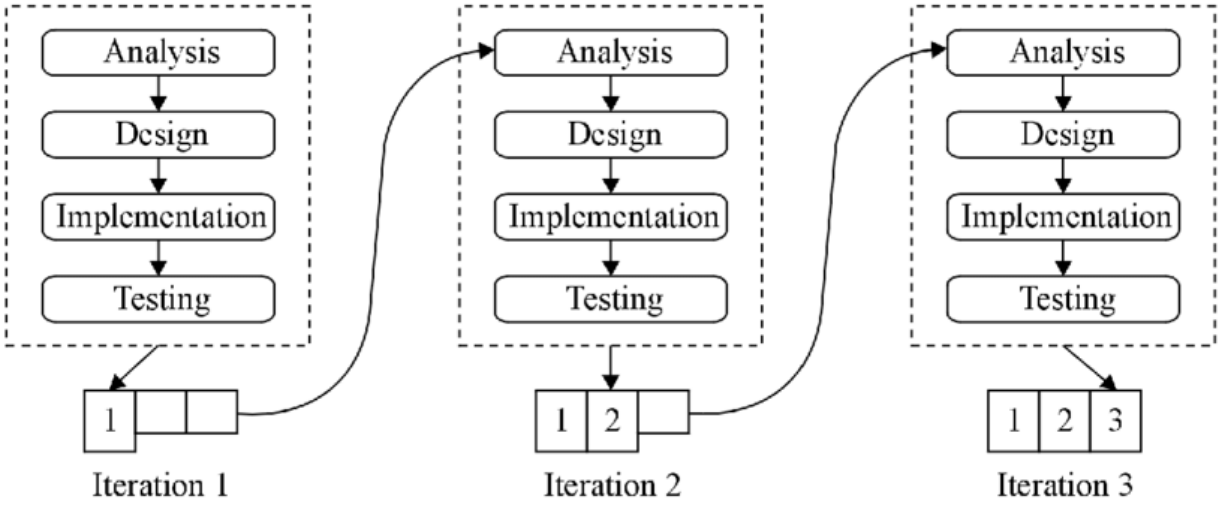


Figure 6: Iterative Incremental Methodology

3.1.1 Planning Phase

The hybrid model for text summarization should be able to summarize text content from pdf document, plain text pasted on the system and text content scrapped from the website URL (Uniform Resource Locators). Extractive text summarization will be conducted first and the output channeled to the abstractive model to generate hybrid summary.

3.1.2 Analysis

In this phase, the specifications of the hybrid model for text summarization will be studied based on the problem that has been identified in planning phase. Analysis will be conducted to choose the best logic, database models and to identify any other requirements. The system architecture is stipulated in this stage. Term Frequency (TF) - Inverse Document Frequency (IDF) and T5 Transformers will be used in the development of hybrid model for text summarization.

3.1.3 Design and Development

The design of hybrid model for text summarization prototype will be produced in this phase. The requirements captured in the previous phases will be used to develop the system. An automatic text summarization model is developed in this phase. The model should have the capability of

summarizing contents extracted from pdf, Wikipedia and from raw text. Python programming language will be used in the model backend development. CSS and HTML will be used for frontend development. Streamlit framework will be used in the frontend development of the model. Streamlit framework helps turn python codes into web apps in very short time for free and no front-end experience required. Building an app is conducted with a few lines of code and simple API calls. Widgets are added easily like declaring variables and no backend code is required to describe routes, handle HTTP web requests, either connect a frontend, draft HTML, CSS and JavaScript. The model will adopt both abstractive and extractive summarization technique.

3.1.3.1 Summary Creation Process

Data Preprocessing will be conducted to convert the data into a machine-readable form of the vector. The process starts with Tokenization of Sentences. The text is divided into sentences .This is implemented via use of a sentence tokenizer from the NLTK toolkit in Python. Once the paragraph is divided into sentences, all special characters and stop words are removed. It's conceivable that the text contains some characters that aren't needed. All of the characters that are not needed will be eliminated. Word Tokenization is conducted using word space where each of the article's phrases will be broken down into words. After word tokenization, each word weighted occurrence frequency is determined and then used to generate extractive summary. The output from the extractive summary is keyed into T5 transformer model so as to generate the hybrid summary.

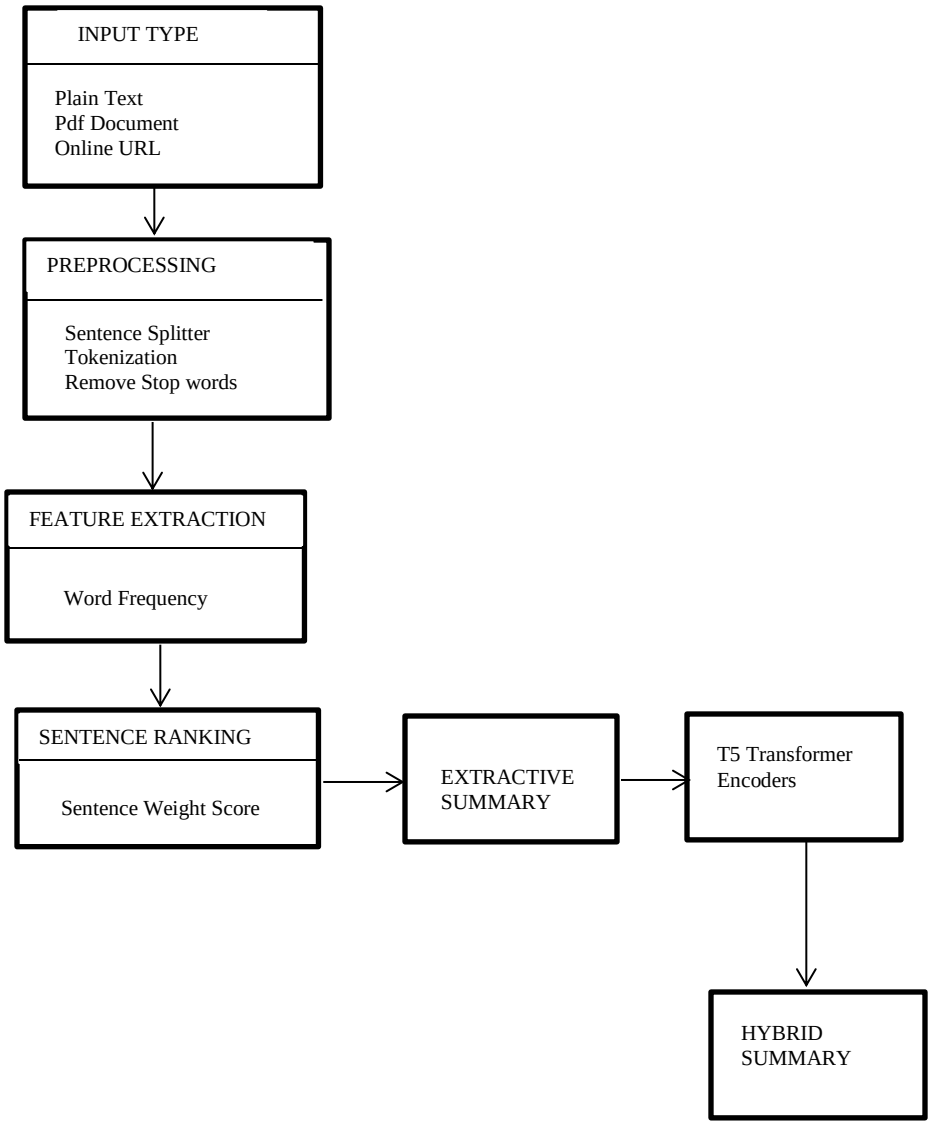


Figure 7 : System Flow

3.1.4 Testing Phase

Testing will begin after the current build iteration has been developed and implemented to find and track any potential defects or problems that may have existed in the the model. The system testing will be carried out in every iteration to determine if all the user requirements are well captured and implemented. The hybrid model for text summarization will be tested to check if it meets the research objective. Plain text will be pasted on the system and submitted to create summary for both abstractive and extractive option. This step should be repeated for pdf content and plain text scrapped from a website URL (Uniform Resource Locators) to make sure that the research objective has been achieved.

3.1.5 Evaluation Phase

The Iterative life cycle ends at this stage. If there are bugs and requirements not met in testing stage, the development is subjected to iterations. If no bugs found, the hybrid model for text summarization is deployed for use. Once the hybrid model for text summarization has passed the testing stage, it is ready for deployment in production.

4 REFERENCE

Aone, C., Okurowski, M. E., & Gorlinsky, J. (2018). Trainable, scalable summarization using robust NLP and machine learning. *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics*, 62–66.

Baxendale, P. (2014). Man-made index for technical literature - an experiment. *I.B.M. Journal of Research and Development, 2*(4).

Boorugu, R., & Ramesh, G. (2020). A Survey on NLP based Text Summarization for Summarizing Product Reviews. *Proceedings of the 2nd International Conference on Inventive Research in Computing Applications, ICIRCA 2020*, 352–356. https://doi.org/10.1109/ICIRCA48905.2020.9183355

Brandow, R., Mitze, K., & Rau, L. . (2014). Automatic condensation of electronic publications by sentence selection. *Information Processing & Management, 31*(2), 675–685.

Celikyilmaz, A., & Hakkani-Tur, D. (2010). A hybrid hierarchical model for multi-document summarization. *ACL 2010 - 48th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference, July*, 815–824.

Chen, Y.-N., Huang, Y., Yeh, C.-F., & Lee, L.-S. (2011). Spoken lecture summarization by random walk over a graph constructed with automatically extracted key terms. *Twelfth Annual Conference of the International Speech Communication Association*.

Chu, W.-S., Song, Y., & Jaimes, A. (2015). Video co-summarization: Video summarization by visual co-occurrence. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 3584–3592.

Dave, H., & Jaswal, S. (2016). Multiple Text Document Summarization System using hybrid Summarization technique. *Proceedings on 2015 1st International Conference on Next Generation Computing Technologies, NGCT 2015, September*, 804–808. https://doi.org/10.1109/NGCT.2015.7375231

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2018). *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding.*

Eberts, M., Ulges, A., & Schwanecke, U. (2015). Amigo-automatic indexing of lecture footage. *2015 13th International Conference on Document Analysis and Recognition (ICDAR)*, 1206–1210.

Edmundson, H. P. (2016). New Methods in Automatic Extracting. *Journal of the ACM (JACM), 16*, 264–285.

Garg, R., Hassan, E., & Chaudhury, S. (2015). Document indexing framework for retrieval of degraded document images. *2015 13th International Conference on Document Analysis and Recognition (ICDAR)*, 1261–1265.

Goyal, P., Behera, L., & McGinnity, T. M. (2018). A context-based word indexing model for document summarization. *IEEE Transactions on Knowledge and Data Engineering*, *25*(8), 1693–1705. https://doi.org/10.1109/TKDE.2012.114

Gupta, A., Chugh, D., Anjum, & Katarya, R. (2021). *Automated News Summarization Using Transformers*. http://arxiv.org/abs/2108.01064

Hassel, M. (2007). *Resource Lean and Portable Automatic Text Summarization*.

Hovy, E., & Lin, C.-Y. (2013). Automated text summarization and the SUMMARIST system. *Proceedings of a Workshop on Held at Baltimore, Maryland*, 197–214.

Kostadinov, S. (2019). Understanding encoder-decoder sequence to sequence model. *Towards Data Science, Https://Towardsdatascience. Com/Understandingencoder-Decoder-Sequence-to-Sequence-Model-679e04af4346*.

Kupiec, J., Pedersen, J., & Chen, F. (2015). A Trainable Document Summarizer. *Proceedings of the 18th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, *2*, 68–73.

Lewis, M., Liu, Y., Goyal, N., Ghazvininejad, M., Mohamed, A., Levy, O., Stoyanov, V., & Zettlemoyer, L. (2019). *BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension.*

Li, K., Wang, J., Wang, H., & Dai, Q. (2014). Structuring lecture videos by automatic projection screen localization and analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *37*(6), 1233–1246.

Liao, P., Zhang, C., Chen, X., & Zhou, X. (2020). Improving Abstractive Text Summarization with History Aggregation. *Proceedings of the International Joint Conference on Neural Networks*. https://doi.org/10.1109/IJCNN48605.2020.9207502

Mallick, C., Das, A. K., Dutta, M., Das, A. K., & Sarkar, A. (2019). Graph-Based Text Summarization Using Modified TextRank. In J. Nayak, A. Abraham, B. M. Krishna, G. T. Chandra Sekhar, & A. K. Das (Eds.), *Soft Computing in Data Analytics* (pp. 137–146). Springer Singapore.

Maxime. (2019). *What is a Transformer?* Inside Machine Learning. https://medium.com/inside-machine-learning/what-is-a-transformer-d07dd1fbec04

Meena, S. M., Ramkumar, M. P., Asmitha, R. E., & Emil Selvan, G. S. (2020). Text Summarization Using Text Frequency Ranking Sentence Prediction. *4th International Conference on Computer, Communication and Signal Processing, ICCCSP 2020*, 0–4. https://doi.org/10.1109/ICCCSP49186.2020.9315203

Mihalcea, R. (2004). *Graph-based ranking algorithms for sentence extraction, applied to text summarization*. *4*, 20-es. https://doi.org/10.3115/1219044.1219064

MuraliKrishna, V. R., Pavan, Kumar, Y. S., & Satyananda, R. C. (2013). A Hybrid Method for

Query based Automatic Summarization System. *International Journal of Computer Applications*, *68*(6), 39–43. https://doi.org/10.5120/11587-6925

Nallapati, R., Zhou, B., Gulcehre, C., Xiang, B., & others. (2016). Abstractive text summarization using sequence-to-sequence rnns and beyond. *ArXiv Preprint ArXiv:1602.06023*.

Nguyen, H., Santos, E., & Russell, J. (2019). Evaluation of the impact of user-cognitive styles on the assessment of text summarization. *IEEE Transactions on Systems, Man, and Cybernetics Part A:Systems and Humans*, *41*(6), 1038–1051. https://doi.org/10.1109/TSMCA.2011.2116001

Patil, M. S., Bewoor, M. S., & Patil, S. H. (2014). A Hybrid Approach for Extractive Document Summarization Using Machine Learning and Clustering Technique. *International Journal of Computer Science & Information Technologies*, *5*(2), 1584–1586.

Pollock, J., & Zamora, A. (2017). Automatic Abstracting Research at Chemical Abstracts Service. *Journal of Chemical Information and Computer Sciences*, *15*(4), 226–232.

Prateek, J. (2019). *How do Transformers Work in NLP? A Guide to the Latest State-of-the-Art Models*. https://www.analyticsvidhya.com/blog/2019/06/understanding-transformers-nlp-state-of-the-art-models/

Radev, D. ., & Erkan, G. (2015). Proceedings of Document Understanding Conference Workshop. *The University of Michigan at Duc*, 120–127.

Rani, S. S., Sreejith, K., & Sanker, A. (2017). A hybrid approach for automatic document summarization. *2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, 663–669. https://doi.org/10.1109/ICACCI.2017.8125917

Shimada, A., Okubo, F., Yin, C., & Ogata, H. (2018). Automatic Summarization of Lecture Slides for Enhanced Student Preview-Technical Report and User Study. *IEEE Transactions on Learning Technologies*, *11*(2), 165–178. https://doi.org/10.1109/TLT.2017.2682086

Song, S., Huang, H., & Ruan, T. (2019). Abstractive text summarization using LSTM-CNN based deep learning. *Multimedia Tools and Applications*, *78*(1), 857–875.

Tan, P. A. (2022). *Introduction to Text Summarization with ROUGE Scores*. Towards Data Science. https://towardsdatascience.com/introduction-to-text-summarization-with-rouge-scores-84140c64b471

Teufel, S., & Moens, M. (2015). Sentence Extraction as a Classification Task. *Proceedings of the ACL/EACL-97 Workshop on Intelligent Scalable Text Summarization*, 58–65.

Tsuchiya, G. (2018). Postmortem Angiographic Studies on the Intercoronary Arterial Anastomoses.: Report I. Studies on Intercoronary Arterial Anastomoses in Adult Human Hearts and the Influence on the Anastomoses of Strictures of the Coronary Arteries. *Japanese Circulation Journal*, *34*(12), 1213–1220. https://doi.org/10.1253/jcj.34.1213

Wei, P., Zhao, J., & Mao, W. (2021). A Graph-to-Sequence Learning Framework for Summarizing Opinionated Texts. *IEEE/ACM Transactions on Audio Speech and Language Processing*, *29*, 1650–1660. https://doi.org/10.1109/TASLP.2021.3071667

Wikipedia. (2022). *Heuristic*. https://en.wikipedia.org/wiki/Heuristic

Yao, K., Zhang, L., Du, D., Luo, T., Tao, L., & Wu, Y. (2018). Dual Encoding for Abstractive Text Summarization. *IEEE Transactions on Cybernetics*, *50*(3), 985–996. https://doi.org/10.1109/TCYB.2018.2876317

Zhang, J., Zhao, Y., Saleh, M., & Liu, P. J. (2020). PEGASUS: Pre-Training with extracted gap-sentences for abstractive summarization. *37th International Conference on Machine Learning, ICML 2020*, *PartF16814*, 11265–11276.

Zhuang, F., Qi, Z., Duan, K., Xi, D., Zhu, Y., Zhu, H., Xiong, H., & He, Q. (2021). A Comprehensive Survey on Transfer Learning. *Proceedings of the IEEE*, *109*(1), 43–76. https://doi.org/10.1109/JPROC.2020.3004555

APPENDICES

Appendix I: Gantt chart

	May			June				July				August			
	Week			Week				Week				Week			
ACTIVITY	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
Topic Selection															
Proposal Writing															
Proposal Presentation															
Requirement analysis															
System Design															
Implementation															
System Testing															
Documentation															
Project Presentation															

Appendix II: Project Budget

NO	ITEMS	AMOUNT (KSHS)
1.	Purchase of Laptop	35,000
2.	Internet charges	5,000
3.	Transport Cost	6,000
4.	Stationery	4,000
6.	Printing, photocopying and binding	2,500
7.	Publication of research paper	8,000
	TOTAL	**60,500**