

**TOPIC: AN IMPROVED AUTOMATIC ARTICLE SUMMARIZATION SYSTEM
USING TEXT RANK**

TABLE OF CONTENTS

DECLARATION

DEDICATION

ACKNOWLEDGEMENT

ABSTRACT

TABLE OF FIGURES

CHAPTER ONE

GENERAL INTRODUCTION AND SUMMARY

1.1 INTRODUCTION

1.2 FIELD AND SUBJECT OF STUDY

1.3 OBJECTIVES

1.3.1 General Objective

1.3.2 Specific Objective

1.4 PROBLEM STATEMENT

1.5 SIGNIFICANCE OF THE STUDY

1.6 BACKGROUND AND JUSTIFICATION OF THE STUDY

1.7 LIMITATIONS OF THE STUDY

1.8 CHAPTER LAYOUT

CHAPTER TWO

LITERATURE REVIEW

2.1 INTRODUCTION

2.2 HISTORICAL DEVELOPMENT OF AUTOMATIC SUMMARIZATION

2.3 DEFINITION OF TERMS

2.3.1 Article

2.3.2 Summarization

2.3.3 Automatic summary

2.3.4 Algorithm

2.3.5 Text Rank Algorithm

2.3.6 Interface

2.3.7 Cluster

2.3.8 Data mining

2.3.9 Extractive method

2.4 CONCEPT OF AUTOMATIC SUMMARIZATION

2.4.1 The Impact of Context in Summarization

2.5 TECHNIQUES FOR ACHIEVING AUTOMATIC SUMMARIZATION

2.5.1 Abstractive Approach Technique

2.5.2 Extractive Approach Technique

2.6 EMPIRICAL REVIEW

2.7 THEORETICAL REVIEW

2.8 RELEVANCY OF RESEARCH

2.9 RELATED METHODS

2.10 LIMITATIONS AND GAPS

2.11 CONCLUSION

CHAPTER THREE

RESEARCH METHODOLOGY

3.1 INTRODUCTION

3.2 DESCRIPTION OF THE PROPOSED SYSTEM

3.3 METHODOLOGY

3.4 EXTRACTIVE SUMMARIZATION

3.5 GENERAL REQUIREMENTS

3.5.1 Functional Requirements for An Automatic Text Summary system

3.5.1.1 Input Design

3.5.1.2 Output Design

3.5.1.3 Logical Design

3.5.2 Non-Functional Requirements for An Automatic Text Summary system

- 3.5.2.1 Software Requirement Specification
 - 3.5.2.2 Hardware Requirement Specification
- 3.6 SYSTEM ANALYSIS MATERIALS TO BE USED
 - 3.6.1 Interview as a method of data Collection
- 3.7 FRAMEWORK OF THE SYSTEM
 - 3.7.1 High-level system framework for previous text summarization
 - 3.7.2 Proposed framework for article summarization
- 3.8 CONCLUSION

CHAPTER FOUR

SYSTEM DESIGN AND DESIGN

- 4.1 INTRODUCTION
- 4.2 SYSTEM DESIGN
 - 4.2.1 Feasibility Study
 - 4.2.2 Technical Feasibility
 - 4.2.3 Economic Feasibility
 - 4.2.4 Operational Feasibility
 - 4.2.5 Schedule Feasibility
 - 4.2.6 Resource Feasibility
- 4.3 MACHINE LEARNING
- 4.4 DESIGN AND DEVELOPMENT PROCESS OF THE SYSTEM
 - 4.4.1 Understanding the problem
 - 4.4.2 Design and Development
 - 4.4.3 Testing and Evaluation
 - 4.4.4 Implementation and Integration
 - 4.4.5 Maintenance
- 4.5 SYSTEM MODELLING
 - 4.5.1 Use Case Diagram
 - 4.5.2 System Testing
 - 4.5.3 Software Testing Strategies
 - 4.5.4 CONCLUSION

CHAPTER FIVE

FINDINGS, CONCLUSION AND RECOMMENDATIONS

5.1 INTRODUCTION

5.2 FUTURE WORKS

5.3 CONCLUSION

5.4 RECOMMENDATIONS

REFERENCES

APPENDIX

CHAPTER ONE

GENERAL INTRODUCTION AND SUMMARY

1.1 INTRODUCTION

Automatic text summarization is a feature of machine learning, natural language processing (NLP), and data mining. As data volumes grow and the need to handle them more efficiently grows, it is becoming a popular study topic. The goal is to discover the core of the supplied text set and reduce its size while covering the key concepts and general meaning and avoiding repetition. Although there is growing interest in summarization and new approaches are constantly being developed, there are still unanswered questions about how to achieve a deeper understanding of the document topic (natural language understanding, NLU), how to handle long documents, and how to improve evaluation methods. Answering these questions will help to further research in this area. In this thesis, we will first learn about the field of text summarizing by studying its history, applications, and evaluation criteria. Then we study and compare various ways from past research on automatic text summarization. We'll start with the simplest and oldest approaches, which rely on simple heuristics, then progress through latent semantic analysis and rhetorical structure theory to neural network-based approaches that are on the rise. Because we are interested in their ability to summarize Finnish literature, we evaluate methods based on their language independence. Lastly, using Finnish news articles, we construct and evaluate an unsupervised extractive text summarizer. We pay special attention to the quality of the data in order to identify its potential for future research.

1.2 FIELD AND SUBJECT OF STUDY

The field of study with regard to this project is Computer Engineering, whereas the subject area is AN IMPROVED AUTOMATIC ARTICLE SUMMARIZATION SYSTEM.

1.3 OBJECTIVES

1.3.1 General Objective

The aim of this project is to create an automatic summarization system that allows users to request from a pool of data and summarize it and also have the ability to upload

documents and get a summarized version of it. It is also aimed at allowing users to be able to discover, consume and digest relevant information faster.

1.3.2 Specific Objectives

- To help users know how to discern the most important ideas in a text and how to ignore irrelevant information
- Help users in integrating central ideas in a meaningful way
- To create and develop an automatic article summarization system using Html, CSS, JavaScript, and Python.
- To give users the ability to upload data and extract the summarized version of the uploaded data.
- To validate and evaluate the performance of the system research.

1.4 PROBLEM STATEMENT

The internet is now accessible to the general people via a variety of devices such as smartphones and smartwatches. As a result, a wealth of knowledge is now accessible via the internet. More information on the internet may make it difficult to select only important information from large documents. A manual summary of information is a complex and time-consuming task due to the content.

1.5 SIGNIFICANCE OF THE STUDY

The significance of this project is to help users extract summarized data from a large pool of data without having to read through a very large number of articles. It allows users to read fewer data but still receive the most relevant information to make a solid conclusion.

1.6 BACKGROUND AND JUSTIFICATION OF THE STUDY

As the web expands, people are becoming overwhelmed by the volume of digitized material and archives. This accessibility has necessitated extensive investigation in the area of the planned content outline. An outline is a book that is derived from at least one record in the source report that illustrates relevant data and is no greater than half the length of the source archive. The programmed rundown is the regular language preparation of AI and data

mining, and the basic idea of the outline is to discover bits of information that comprise the data of the entire source archive. Such systems are widely used in business today; web search tools are one example, and others include a record summary, image collection, and recordings. It has become clear that this examination should and must be prioritized, which is why this research is being conducted. Everyone has had the task of condensing an article or record so as to make short and brief data from the entire article or archive given, this procedure supposedly is a distressing errand, as you begin to stress over which sentences are significant, what key expressions should not be surrendered, or more all, the rundown should convey all the significant data required without composing back the entire article.

The task of the programmed article synopsis framework is to do away with the aforementioned concerns. The programmed outline framework accomplishes this by taking a single article or groups of articles and giving a concise and familiar rundown of the most important data. Throughout the years, the synopsis framework has been utilized to help individuals in various sectors. As defective as they look to be, they have just appeared to help clients and moreover give an approach to other programmed outline applications. This task is to provide a programmed article outline framework that will also allow clients to transfer files, and research articles, and obtain a summarized version of it. The extractive and abstractive approaches are used for automatic summarization. The extractive methodology involves selecting sentences from the text that best speak to its outline. Because of its origins in the 1950s, extractive outlining processes have been widely used for a long time. Instead of attempting to comprehend the substance of the material, it is more about determining how to comprehend the significance of each sentence and its relationships with one another. The Abstractive synopsis, on the other hand, is associated with striving to comprehend the substance of the content and after that presenting an outline based on that, which may have indistinguishable sentences from the ones in the first content.

Abstractive outline attempts to make its own sentences and is unquestionably a stage towards progressively human-like synopsis. Over the years the summarization system has been built to help people in various fields, imperfect as they are, they have already shown to be very useful to users and also give way to other automatic summarization applications.

In the early days, text summarization was done exclusively using rule-based algorithms. It was called “importance evaluator”, this was worked based on ranking different parts of a text

according to their importance. Two important knowledge bases were used by the evaluator; one of them being the “importance rule base” which made use of the ‘IF-THEN’ rules and the other being the “encyclopedia” which contained domain-specific world knowledge represented using a network of frames. The essential rule-based method employs a theory known as Hierarchical Propositional Network (HPN), in which numerical representations are assigned to the conceptual units of extended linear representations (ELR) of sentences. The goal interpreter then uses a referential framework to interpret the significance of each sentence. An approach known as the "Production rule system for summarization" was used in 1984. It essentially operates in three steps: a) inference, b) scoring the format rows for importance, and c) selecting the right ones as a summary. As academics in the field of summarization techniques advanced, several advances were made in interpreting the significance of sentences in a textual data corpus.

1.7 LIMITATIONS OF THE STUDY

Because text summarization gives the condensed subtext of the original text, there are always a lot of obstacles in determining whether or not we acquire the appropriate text summary from the enormous document. Whether or whether the summary contains essential sentences and terms. The following are some of the limitations of the study:

Extract hidden semantic relationship:

There are several sentences in the text that are related to one another. There are numerous sentences in the text. Some sentences in the document are related to each other based on semantic links between textual concepts. Many sentences are related to one another and depict various aspects of the text material. As a result, summarizing such sentences in the summary is usually a difficult challenge.

Relevance detection;

While creating a text summary, it is also necessary to locate relevant sentences in the text documents so that they can be chosen to build up the final summaries. Finding the important sentences in the book to include in the final summary is usually a difficult undertaking. Because it has a significant impact on the quality of our summary. The "Code Quality Principle" can be utilized to identify key sentences in our text composition. To assure the

relevancy of the sentence, many factors such as sentence position within the text, word and phrase frequencies, and title overlap are used.

1.8 CHAPTER LAYOUT

This project is divided into five (5) main chapters which are Introduction, Literature Review, Methodology, System Analysis & Testing, and Conclusion & Recommendation. Chapter 1 is the General Introduction and Summary. This chapter has discussed the overview of trending events underlying the topic and area of study, specifically, a brief introduction about Automatic summary as well the problem statement, objectives, and scope of the work of the intended research and design work. Chapter 2 provides a Literature review of the related works done by previous researchers and studies related to this project. Chapter 3 specifies the Methodology which discusses the ways or methods used to conduct this project. Also, all the procedures used to conduct the experiments will also be mentioned here. Chapter 4, the Implementation & Results and Analysis will discuss all the important requirements needed to conduct the experiments. This chapter also contains the result and the analysis of the system. Finally, Chapter 5 is the Discussion, Conclusion and Recommendations that concludes and discusses recommendations for future research.

CHAPTER TWO

LITERATURE REVIEW

2.1 INTRODUCTION

The purpose of this chapter is to discuss and examine relevant literature in relation to the research on the Enhanced Automated Article Summarization System Using Text Ranks. The description, classification, and comparison of earlier research studies and articles, as well as observations and conclusions concerning their benefits, limitations, and suggestions, will be used to conduct a critical analysis of distinct segments of a published body of knowledge. The literature review for this study is focused on the functionality, processes, and components of the evaluated systems. While picking a system for systematic review and comparison among the selected systems, the main focus is on the technology utilized and the functionality supplied by the system. The goal of this literature review is to investigate the requirement definition, limitations, and strengths of various systems. After reviewing these systems, insights can be learned and incorporated into the suggested system to overcome the limitations of this type of technology. Aside from that, functions that are commonly used by the selected systems could be chosen as functions of the proposed system as well. The technology employed by the chosen systems is also investigated in order to improve the suggested system.

2.2 HISTORICAL DEVELOPMENT OF AUTOMATIC SUMMARIZATION

Luhn pioneered automatic summarization in the 1950s, employing sentence extraction techniques and executing his methodology on specialist papers and magazine articles. This fundamental idea that Luhn advanced subsequently established the future research of programmed rundown, specifically that a few phrases in an archive are visual and the sentences that draw out the most substantial data in the archive are those that have numerous such engaging words near one another. Luhn proceeded to acquaint a technique with remove remarkable sentences from the content utilizing highlights, for example, word and expression recurrence, he proposed to gauge the sentences of a report as an element of high thickness normal words, and further went to depict an example based on key expressions nouns and verbs, and further went to depict an example based on key expressions nouns and verbs, and further went to depict an example based on key expressions nouns;

- Cue Method: in this technique, the pertinence of a sentence is determined dependent on the existence of a specific sign word in the prompt lexicon.
- Title Method: The heaviness of a sentence is figured as an entirety of all substance words showing up in the title and heading of a book.
- Location Method: This technique accepts that sentences showing up at the start of individual passages have a higher likelihood of being significant.

Several studies have been published throughout the years to address the challenges of automatic summarization, with novel ways being developed with the advancement of modern technologies like as machine learning, natural language processing, and data mining.

2.3 DEFINITION OF TERMS

- 2.3.1 Article: Article is a piece of writing included with others in a newspaper, magazine or other publications. An article can be an essay, reports, accounts, column and composition etc.
- 2.3.2 Summarization: Summarization can be described as the demonstration of communicating the most significant reality or thought regarding a person or thing in a brief and familiar structure.
- 2.3.3 Automatic summarization: The process of shortening a text document with software in order to get the major points of the original documents.
- 2.3.4 Algorithm: A process to be followed in order to solve a problem.
- 2.3.5 Text Rank Algorithm: An automatic summarization technique that ranks text chunks in order of their importance in the text document.
- 2.3.6 Interface: The shared boundary between users and the computer.
- 2.3.7 Cluster: A group of similar things/occurring closely together.
- 2.3.8 Data mining: Practice of examining large pre-existing databases in order to generate new information.
- 2.3.9 Extractive method: Involves the selection of phrases and sentences from the source document to make up the new summary.

2.4 CONCEPT OF AUTOMATIC SUMMARIZATION

An automatic summarizing system takes a large number of documents as input and attempts to generate a succinct and comprehensive summary of the most essential information in the input. Creating a brief and comprehensive summary necessitates the capacity to recognize, change, and join information expressed in various sentences in the input. All of these considerations go into the design of the automatic summarization system.

2.4.1 The Impact of Context in Summarization:

Synopsis structures regularly have additional confirmation they can utilize in order to decide the most noteworthy purposes of the document(s). For example, when delineating locales, there are discourses or comments coming after the blog section that are extraordinary wellsprings of information to make sense of which parts of the blog are essential and fascinating. In a scientific paper summary, there is a great deal of information, for instance, referred to papers and assembling information which can be used to perceive noteworthy sentences in the principal paper. In the going with, we portray some of the settings in a more nuanced.

- Web Summarization:

Pages contain numerous components that cannot be omitted, such as images. Because the printed data they have is typically scarce, implementing content rundown tactics is limited. In any case, we can think of the setting of a web page, for example, snippets of data separated from the core of the significant number of pages linking to it, as extra material to improve the outline. The first thing they check into in this method is where they query web search tools and bring the pages with links to the defined site page. At that point, they break down the up-and-coming pages and heuristically select the finest sentences containing linkages to the page. Delort et al. (2003) extended and improved on this process by employing a formula that attempts to select a sentence about a similar topic that spans as many areas of the web page as possible. Propose a system for blog synopsis that first extracts agent words from comments and then selects significant sentences from blog entries containing delegate terms.

- Scientific Articles Summarization:

- Finding various papers that refer to the target paper and concentrating the phrases where the references appear to distinguish the major sections of the objective document is a vital wellspring of data while abridging a scientific publication (for example, reference-based rundown). [Amjhad \(2011\)](#). suggest a language model in which each word in the reference setting phrases is assigned a likelihood. They then use the KL dissimilarity technique to rate the relevance of sentences in the first publication (for example finding the comparability between a sentence and the language model).
- Email Summarization:
The email contains some characteristics that demonstrate the components of both spoken communication and written information. For example, like in spoken talks, rundown methods must consider the intelligent notion of the exchange. [Nenkova et al \(1970s\)](#). In this approach, I introduced early research by providing a strategy for producing a rundown for the first two degrees of the string discourse. After some time, a string consists of at least one debate involving at least two individuals. They choose a message from the root message and from each reaction to the root, keeping the cover with the root setting in mind. [Rambow et al \(1970s\)](#). used an AI algorithm and included string-related highlights as well as email structure highlights, for example, phrase position in the track, number of beneficiaries, and so on. [Newman et al \(1970s\)](#). Show a framework for condensing a whole letterbox rather than a single string by grouping messages into thematic groups and then extracting synopses for each group.

2.5 TECHNIQUES FOR ACHIEVING AUTOMATIC SUMMARIZATION

There are two general approaches to achieving an automatic summarization system, which is the Abstractive and Extractive approaches. The two approaches are explained in brief below:

- 2.5.1 Abstractive Approach: The abstractive approach will generate new phrases or use new words that were not in the original text to generate the summary. Naturally, abstractive approaches are harder, as the model has to first understand the documents and then try to express that understanding using new words and phrases to create a perfect summary. The abstractive approach is barely used as it deals with semantic problems and produces

less effective summary than the extractive approach. This method uses advanced natural language processing and deep learning which is a growing field in itself.

2.5.2 Extractive Approach: In this approach the automatic system takes out sentences from the entire collection, without altering the sentences themselves. This approach will check out the important sentences and use these sentences to form the summary. There are different techniques and algorithms that are used to check out the weight of the sentences and then rank those sentences depending on how similar and important they are. They include;

- Text Rank Algorithm: Text Rank is a universally useful chart-based positioning calculation for Natural Language Processing (NLP), it is like PageRank calculation and is an unaided learning approach towards extractive synopsis. The essential thought of Text Rank is to give a score of each sentence in a book and take the top-n sentence and sort them as they show up in the content to fabricate a programmed rundown. A chart is developed by making a vertex for each sentence in the record; the edges between sentences depend on some type of semantic closeness.
- Cosine Similarity: Cosine likeness is a proportion of closeness between two non-zero vectors of an internal item space that estimates the cosine of the edge between them, sentences are spoken to as a pack of vectors so the cosine similitude discovers comparability among sentences.
- Sub-modular set function: In arithmetic, a submodular set capacity is a set capacity whose worth casually has the property that the distinction in the gradual estimation of the capacity that a solitary component makes when added to an information set increments.
- In rundown submodular works normally model ideas of inclusion, data portrayal and decent variety. For instance, in a report outline one might want the rundown to cover exceedingly significant data in the record, this is an occurrence of set spread in submodular capacities.
- Lex Rank Algorithm: Lex rank is a more principled way to estimate sentence importance using random walks and eigenvector centrality, Lex rank is very similar to text rank.

Extractive outlines frequently give preferable outcome over abstractive rundowns, this is on the grounds that abstractive synopsis techniques adapt to issues like semantic portrayal, surmising and characteristic language age which is generally harder than information driven methodologies like sentence extraction.

2.6 EMPIRICAL REVIEW

Machine learning techniques for text summarization:

Machine learning techniques are classified into two types: supervised and unsupervised learning. Supervised learning implies that the machine is learning under the supervision of a human and is being taught with data that has already been checked and labeled as correct. The outcome of this learning strategy is already known and proven. Once the computer has learned from genuine training data, it is given new unknown data to forecast outcomes. It aids in the optimization of various results based on criteria. They are mostly employed to address complex computational issues in the real world. Supervised learning is divided into two techniques: classification technique and regression technique. Using training data, the regression technique generates a single continuous output value. As the name implies, a classification strategy classifies or categorizes input into some output category. Depending on the output categories, classification might be binary or multiclass. In his paper, author Cheng-Zen Yang suggested a supervised learning technique for producing bug report summaries.

Unsupervised learning is a technique in which unlabeled data is presented as input and algorithms are used to generate output. It is used to discover all possible patterns or classifications in data. In this strategy, the model is not supervised. It is further subdivided into two techniques: clustering and association. The clustering technique takes incoming data and, after processing it, divides it into multiple clusters based on common traits. The association technique aids in joining or associating data elements in order to uncover some relationship in a very big data collection or database. In his paper "AUSUM: Approach for Unsupervised Bug Report Summarization," author Senthil Kumar developed a method for producing bug report summaries using an unsupervised learning technique.

Evaluation parameters of generated summary:

- Precision - Precision is defined as the ratio of successfully predicted positive predictions to all predicted positive observations.

$$\text{Precision} = \text{TP} / (\text{TP} + \text{FP})$$

- Recall (Sensitivity) – Recall is the ratio of properly anticipated positive predictions to all observations in its class.

$$\text{Recall} = \text{TP} / (\text{TP} + \text{FN})$$

- F1 score - F1 score is the weighted average of precision and recall. Therefore, this score considers false positives as well as false negatives in to consideration. F1 score is more useful if we have an uneven class distribution.

$$\text{F1 Score} = 2 * (\text{Recall} * \text{Precision}) / (\text{Recall} + \text{Precision}).$$

2.7 THEORETICAL REVIEW

Lin, Chin-Yew (2004). The author introduced Recall-Oriented Understudy for Gisting Evaluation ROUGE in this research. This is an automatic text summarization evaluation program. The study also established four new ROUGE measures: ROUGE-N, ROUGE-L, ROUGE-W, and ROUGE-S. It assesses summary quality by comparing the generated summary to other ideal summaries created by people. These methods are effective for the automatic evaluation of single and multi-document summaries.

Akshil Kumar and colleagues (2017). The author of this study evaluated and compared the performance of three different algorithms. First, the various text summarizing approaches are explained. To extract important keywords for inclusion in the summary, extraction-based techniques are used. Three keyword extraction techniques were employed for comparison: Text Rank, Lex Rank, and Latent Semantic Analysis (LSA). Three algorithms are described and implemented in Python. The ROUGE 1 is used to assess the efficacy of the extracted keywords. The algorithms' output was compared to handwritten summaries to assess performance. Finally, the Text Rank Algorithm outperforms the other two algorithms.

Pankaj Gupta and colleagues (2016). In this study, the author explored various strategies for sentiment analysis and text summarization. Sentiment analysis is a machine learning strategy which machine learns and assess the feelings, emotions existing in the text. Machine learning techniques such as Naive Bayes Classifier and Support Machine Vectors (SVM) are employed. These approaches are used to extract emotions and sentiments from text data, such as movie or product reviews. Natural language processing (NLP) and linguistic aspects of

sentences are employed in text summarizing to check the importance of the words and sentences that might be included in the final summary. In this paper, a survey of prior research work connected to text summarization and sentiment analysis was conducted, so that a new research area can be investigated while considering the benefits and drawbacks of present approaches and tactics.

[2017] Tacho Jo. In this research, the author presented a variant of KNN (K Nearest Neighbor) in which words are supposed to be features of numerical vectors that represent text. The similarity of feature vectors is calculated by taking into account both attribute and value similarity. Text summarization is considered as a categorization task. The text is broken down into paragraphs or sentences. The classifier categorizes each paragraph or sentence as either summary or no summary. The sentences classed as 'summary' are extracted as a result of summarizing the text, while other text is disregarded. The proposed version of KNN produces better results in text classification and clustering. The modified version of KNN leads to a more compact representation of data item and better performance.

2.8 RELEVANCY OF RESEARCH

The amount of information available on the internet is unlimited. Consider a typical college student who must read thousands of pages of documentation each semester. This is why text summary is required. The primary goal of Automatic text summarizing is to extract the most precise and helpful information from a large document while eliminating irrelevant or less important material. Automatic Text summarization can be done manually, which takes time, or automatically using machine algorithms and AIs like this project, which takes much less time and is a better alternative.

2.9 RELATED METHODS

There are numerous approaches to text summarizing research that are applied there. Figure 1 depicts the method distribution for this study from 2008 to 2019. According to literature surveys, fuzzy logic has been the most extensively utilized method in text summary over the last ten years.

The fuzzy logic method is the most popular since it has been used to extract or determine the final value of words or phrases included in the summary in the previous ten text summarizing

studies. Because it incorporates the role of people in examining sentences and reaching agreement on the selection of particular words to form summary sentences, the fuzzy logic approach can prevent data contradiction (Abbasi-ghalehtaki et al., 2016).

The way fuzzy systems work is to use multiple inputs from various features or indices. Various features or indicators to produce a summary, for example, the frequency of words that appear, a similarity with the title, the relevance of words or sentences, sentence length, sentence position, and so on.

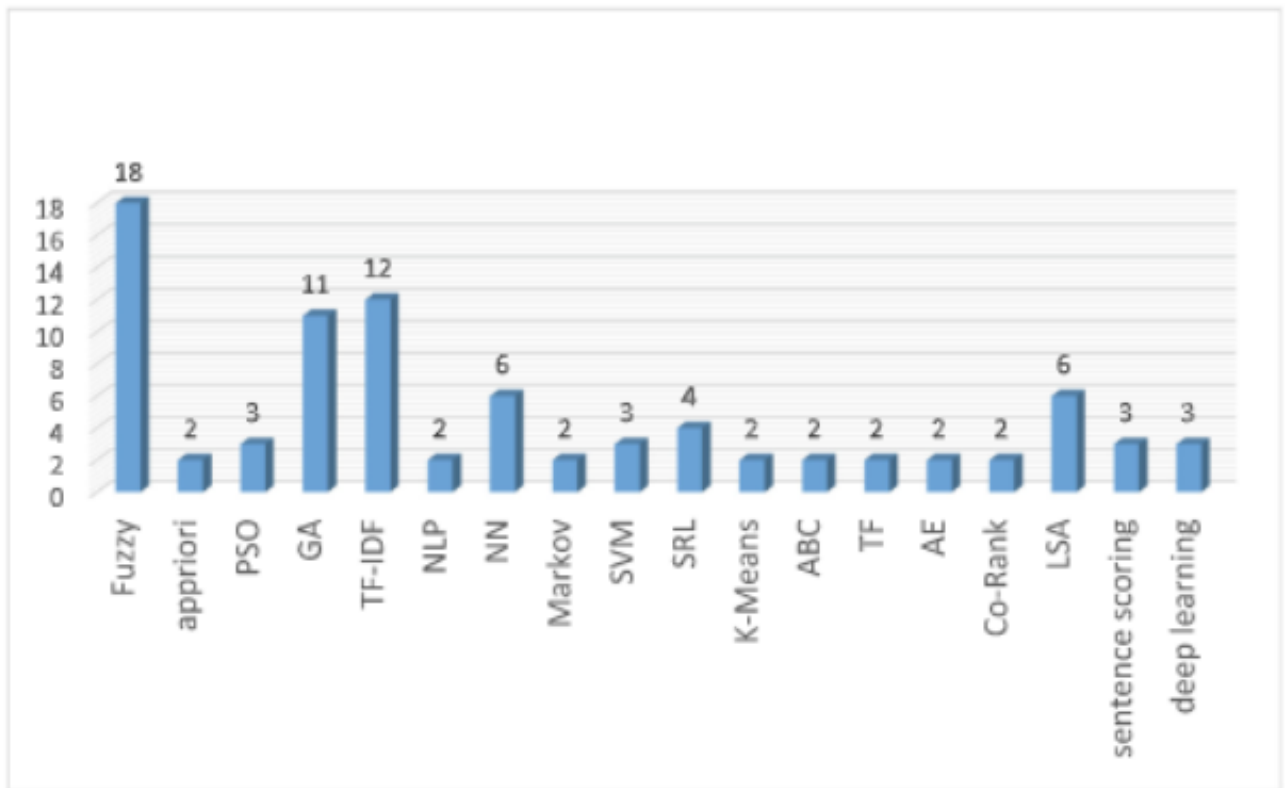


Fig.1. Distribution of Methods used in Automatic text summarization.

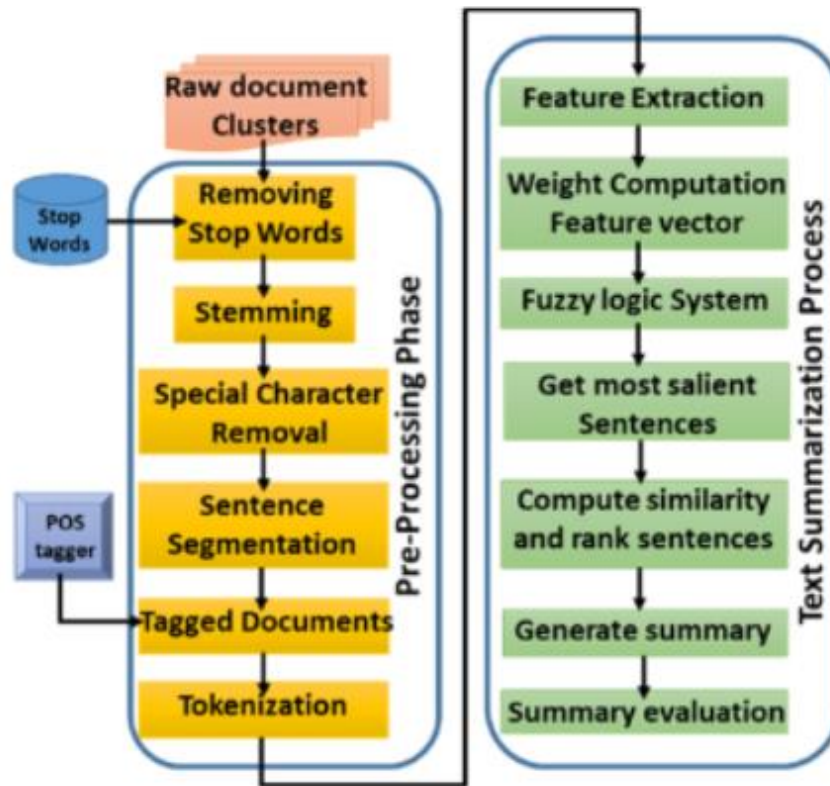


Fig.2. DS System using Fuzzy Logic (Patel et al., 2019).

The fuzzy inference system is then provided the score of each feature as input. Furthermore, human knowledge is leveraged in the form of IF-THEN rules, allowing us to generate more precise phrases capable of producing high-quality summaries (Goularte et al., 2019; Abbasi-ghalehtaki et al., 2016; Babar and Patil, 2015; Patel et al., 2019; Lee et al., 2013). The most recent research employing fuzzy is Goularte's (Goularte et al., 2019) research, which provides extractive summaries. This approach uses a fuzzy assessment of several criteria such as frequency, similarity, position, and sentence length to find the most significant information from the text. This system explores associated features in order to reduce dimensions and hence the number of fuzzy rules. The proposed experiment by Goularte was evaluated using a private dataset of Brazilian Portuguese text supplied by students. This fuzzy summary system is compared against four other summary systems: baseline, score, model, and sentence utilizing precision, recall, and f1/f-measure, as well as CI for F1 with summary sizes of 40%, 30%, and 20%. The fuzzy technique outperforms the state of the art for sizes of 30% and 20%. The fuzzy technique yields precision 0.366, recall 0.496, F1 0.421, and CI for F1 0.389-0.450 with a

summary size of 30%. The fuzzy technique yields a precision of 0.417, recall of 0.398, F-1 of 0.406, and CI for F1 of 0.369-0.436 for a size of 20%. However, at a summary size of 40%, the fuzzy technique is less precise than the model system; the model precision is 0.316, while the fuzzy precision is 0.315. The most recent research employing another fuzzy method to determine keywords is the research of (Patel et al., 2019) by incorporating the TF-IDF method. MDS refers to the employment of a combination of fuzzy and TF-IDF algorithms to summarize numerous documents. This MDS system is depicted in Fig. 2 to aid comprehension. The MDS system outperformed other systems such as the top-ranked DUC 2004 peers, Text Lex An, Item Sum, Baseline, Yago Summarizer, MSSF, and Patsum when tested using the DUC 2004 dataset. Patel et al. (2019) great at measuring recall 0.1555 on Rouge-2 and superior at measuring recall 0.068 and f-measure 0.038 on Rouge-4 using MDS. MDS by Patel et al. (2019) surpassed Parsum in terms of recall with a very small value difference, whereas Parsum excelled in terms of precision.

Lierde's (Lierde and Chow, 2019) research employs fuzzy combining with graphical techniques, specifically Fuzzy hypergraphs. This model was tested on the DUC 2005, DUC 2006, and DUC 2007 datasets to generate extractive summaries. With rigorous evaluation, the findings outperformed the LDA, K-Means, Terms, AC, and DBSCAN approaches. You can broaden the scope of the summary by utilizing this model to select sentences. However, it falls short in terms of summary clarity. The most often used text summarization methods are TF-IDF and LSA. The TF-IDF method employs a statistical approach technique. This method has been used in text summarization research such as (Khan et al., 2019; Azmi and Altmami, 2018; Alami et al., 2019; Alzuhair and Al-dhelaan, 2019), among others. This is one algorithm for analyzing the link between a phrase/sentence and a group of documents. The idea is to compute the TF and IDF values (Robertson, 2004). Each sentence is considered a document for a single document. The frequency of recurrence of the word (t) in the sentence (TF) indicates the importance of the term in the document. While IDF is the number of sentences in which a word (t) appears, it indicates the frequency with which the word is used. Word weight will be higher if it appears frequently in a document and lower if it appears in several documents. The weighting of words in a document is calculated by multiplying the TF value by the IDF value.

CLA, Restricted Boltzmann Machine (RBM), Analytical Hierarchy Process (AHP), AMR, Abstract Meaning Representation (AMR), abstractive summarization (AS), single + dual train, MMR, SOF (WP/POS/NER/WF/HF), Recurrent neural network (RNN), Sentiment Memory (SM), (hierarchies agglomerative clustering) HAC, Multi-Objective Artificial Bee Colony (MOABC) Narrative abstractive summarization, rank-biased precision summarization (RBP-Sum), and the decay topic model are all methods for summarizing narratives (DTM). NATSUM is a brand-new approach among these (Barros et al., 2019). NATSUM is a machine learning-based abstractive summarization method. The NATSUM evaluation scores shine in rouge, grammatical, non-redundancy, and coherence evaluations.

2.10 GAPS AND LIMITATIONS

Many other research works have contributed to the development of an automated article summarization system and the systems have been yielding good results. These related works together with the techniques used and their limitations are listed below;

Author & Year	Article Title	Techniques	Limitation
Collins, (2015)	What Automatic Summarization is all about	Application Area of Automatic summarization	The areas where automatic summarization was applied showed some challenges
Kathleen (August 2014)	A survey of text summarization techniques applying the text Rank algorithm	The Text Rank algorithm	Although the text Rank algorithm can be seen as one of the best algorithms when dealing with text summarization its limitation is grammar, while concatenating it does not fully eradicate

			grammar errors
Amjad, (January 13, 2011)	Coherent citation-based summarization of scientific paper	Topic words, page Rank, Lex Rank	Technique was limited to the use of key phrase, and it couldn't go further to summarize other type of data
John, (march 2011)	Personalized and automatic social summarization of events in videos	Topic words, PageRank	The technique gave lots of issues has it didn't drop some redundant words
Ibrahim F. (January 2010)	Semantic Graph Reduction Approach for Abstractive Text Summarization	Semantic graph-based approach	This technique is limited to single document summarization it cannot be used for multiple document summarization
Atif (February 2008)	A Review on Abstractive Summarization Methods"	Structure based abstractive method (rule based)	The abstractive approach can be seen has the best technique for summarization but it is very much limited as it has to deal with a whole lot of limitation like semantics and the study of advanced natural language processing which in

			itself is still a growing field
--	--	--	------------------------------------

Table 1.

2.11 CONCLUSION

In this chapter, so far, we have seen a review of An Automatic text system, reviews and some other things to which people referred to as Text summary. In the next chapter, we shall look at the system analysis and research methodology.

CHAPTER THREE

RESEARCH METHODOLOGY

3.1 INTRODUCTION

This chapter focuses on the design, development and system modeling. The system to be developed will help in automatic text summarization that is presenting the source document into a shorter version with semantics.

To achieve this aim, we have carried out a proper review of various techniques and algorithms used for summarization.

3.2 DESCRIPTION OF THE PROPOSED SYSTEM

Summarization as we know is the act of writing concise and fluent summary from a document or multiple documents. Since time immemorial summarization has been done majorly by hand i.e., humans, the problem with this system is that humans are very prone to errors and the possibility of leaving out the most important sentences that should be included in the summary compared to a computer gene. The system at a high level of abstraction simply allows the visitors to the site view summary of previously uploaded documents by signed up users. This implements text summarization to automatically create an abstract of the uploaded document which the visitors to the website can read. Interested visitors may want to get the full document. This is only possible for signed up users hence the system compels such visitors to sign up and hence has the privileges of other signed up users such as ability to download full document and ability to upload your own documents. A handy tool for signed up users is also an ability to automatically create your document abstract using the summarization engine embedded within the system.

3.3 METHODOLOGY USED

The Adopted methodology for the proposed system is Agile. Agile software development is a collection of iterative software development approaches in which requirements and solutions emerge through collaboration among self-organizing cross-functional teams. In

general, agile methods and processes promote a disciplined project management process that encourages frequent inspection and adaptation, a leadership philosophy that promotes teamwork, self-organization, and accountability, a set of engineering best practices designed to allow for the rapid delivery of high-quality software, and a business approach that aligns development with customer needs and company goals. Agile Methodology refers to a practice that encourages continuous iteration of development and testing throughout the project's software development lifecycle. Unlike the Waterfall model, both development and testing activities are carried out concurrently in the Agile model of software testing. One of the most easy and efficient approaches for transforming a vision for a business need into software solutions is the Agile software development process. Continuous planning, learning, improvement, team communication, evolutionary development, and early delivery are all used in agile software development methodologies. It encourages flexible reactions to change.

Four values that are emphasized in agile development.

- Interactions between individuals and teams regarding processes and tools
- Working software trumps extensive documentation.
- Customer involvement in contract negotiations
- Sticking to a plan by Responding to change

Advantages of Agile Methodology

- It is flexible and adaptable
- Lower Cost
- Improved Quality
- End-user satisfaction

Disadvantages of Agile Methodology

- Scalability
- Skill Require

3.4 EXTRACTIVE SUMMARIZATION USING TEXT TRANK

The extractive approach of building an automatic summarization system takes several

sentences or phrases and stacks them to produce a summary. Text Rank algorithm is an extractive and unsupervised text summarization technique. To generate the required summary the following steps would be considered using Text Rank algorithm:

- All the texts contained in an article will be concatenated.
- The texts will be split into individual sentences.
- Vector representation for each sentence will be found
- Similarities between sentence vectors are then calculated and stored in a matrix
- The similarity matrix is then converted into a graph for sentence rank calculation
- Finally, a certain number of top ranked sentences form the final summary

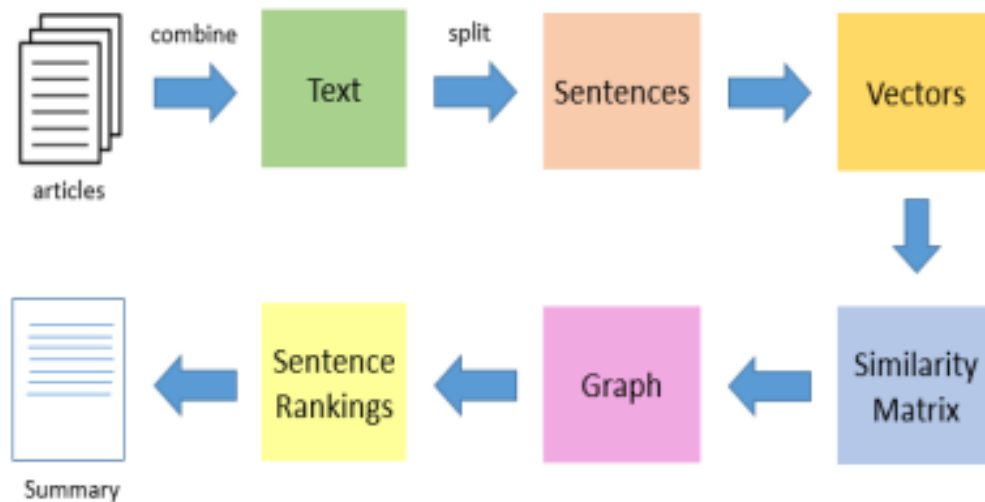


Fig 4.

3.5 GENERAL REQUIREMENTS

In order to implement an efficient and effective system one needs to understand the business and technical requirements of the implementation process. The main requirement of the proposed system is categorized into:

- Functional requirement
- Non-functional requirement
- Software Requirement Specification
- Hardware Requirement Specification

3.5.1 Functional Requirements for An Automatic Text Summary system

Functional requirements are system features that developers must include in order for users of the system to achieve their objectives. They define the fundamental system behavior under certain conditions

3.5.1.1 Input Design

User-oriented inputs are transformed into a computer-based system format in the input design. The design of menus and prompts is a key component of input design. The user's options are predefined for each possibility. Logical flow, data storage, source, and destination are shown in the data flow diagram. The gathering and grouping of input data into groups of related data.

3.5.1.2 Output Design

The goal of the output design is to show output on a screen in a specific format. The capabilities of the output device, the necessary response times, etc. are taken into account. In order to provide users with information that is clear, precise, and quick, the form design elaborates on how output is displayed and the layout accessible for information capture.

3.5.1.3 Logical Design

The logically suggested data is what logical data design is all about. It is possible to construct a form so that the meaning may be understood by each and every piece of data. A clear knowledge and notion of the associated data required to build a form and process information and events should be provided by logical data design.

Few functional requirements of the system shall perform;

- The system shall allow students to view their current academic performance
- The system shall generate reports for each student's performance.
- The system shall allow students to input necessary details for prediction

3.5.2 Non-Functional Requirements for An Automatic Text Summary system

Non-functional requirements are descriptions of how the system should operate; they list all current requirements that the functional requirement does not address. These nonfunctional criteria include things like;

- Usability: The simplicity of this system's operation is its most crucial criterion. The system is designed in such a manner that users who are unfamiliar with the technical details of computer systems would find it easy to use and comfortable.
- Maintainability: The suggested solution is highly adaptable to new trends and platform development.
- Interoperability: This is the application's capacity to work with many versions of Windows, including Windows 7, Linux, and MAC, which are the most recent trends in computer system devices.

Few non-functional requirements of the system shall perform;

- The system should present the required feature at any given time.
- The system should present text using readable font style i.e., no demographic font.
- The system should generate the summary in less than 5 second with an internet speed greater or equal to 200kbps.
- The system should be accessible to end user at any location in the world. The system should be available at every hour of the day.
- The system should be responsive to any user device.
- The system's functionalities should work on every recent browser.

3.5.2.1 Software Requirement Specification

The developed system is not platform dependent in order to effectively solve the problem that led to the development of this project. As a result, it can run on any platform. The reviews of the developed system's specifications are listed below;

Operating System	Windows OS, Linux, and Mac OS (Python must be installed on the selected Operating System used.)
Front-End Technologies	HTML5, CSS3, JavaScript
Server-side Technologies	Python3, Flask
Server	Python3
Back-End Tools	Visual studio code

Table 2 Software Requirement

3.5.2.2 Hardware Requirement Specification

Devices	Desktop Devices, Laptop
Processor	x64 Processor: AMD Opteron, AMD Athlon 64, Intel Xeon with Intel EM64T support, Intel Pentium IV with EM64T support or higher
Memory	Minimum of 1gb Ram
Storage	Minimum 1gb

Table 3. Hardware Requirement

3.6 SYSTEM ANALYSIS MATERIALS TO BE USED

System analysis is the process of acquiring and evaluating facts, detecting problems, and recommending system changes. System analysis is a problem-solving activity that necessitates frequent interaction between system users and system developers (Jawahar, 2012). A critical phase of any system development process is system analysis or study. The system is seen as a whole, the inputs are identified, and the system is thoroughly examined to find issue areas. The solutions are presented as a suggestion. On user request, the proposal is assessed and appropriate revisions are made. It is a comprehensive assessment of end-user information demands that results in functional requirements that serve as the foundation for the design of a new information system. System analysis is actually performed in order to subsequently perform a system design.

3.6.1 Interview as a method of data Collection

Interviews are used to collect data from small groups on a wide range of topics. You can use structured or unstructured interviews. Structured interviews are similar to surveys, with the same questions asked in the same order for each topic and multiple-choice answers. In unstructured interviews, questions vary by topic, may depend on answers to previous questions, and do not have fixed answer options. Where face-to-face interviews cannot be conducted, study participants/respondents may be interviewed using video conferencing tools in

addition to regular telephone calls. You can safely use the different video conferencing tools such as: Microsoft Teams, Google meet etc. The main purpose of interviews as a data collection tool is to collect data comprehensively and centrally. As Pauline Young pointed out, the purpose of the interviews could be the exchange of ideas and experiences, the collection of information related to a very wide range of data in which respondents repeat the past and define their present and future desires. there is. discuss options.

Importance of Interview:

- It is the most suitable method for assessing personal qualities.
- It has clear value for diagnosing and treating emotional problems.
- This is one of the essential foundations for implementing the recommended procedures. It provides information that complements other data collection methods.
- It can be used not only for observation but also for verifying information obtained by the response method.

Types of Interviews / questions asked

users	Question asked
Students	<p>How do you feel when reading a long article?</p> <p>How do you summarize your thesis?</p> <p>What do you think about a summary tool?</p> <p>What do you when you can't summarize your thesis?</p>
Article writers	<p>What solution do you attain to when having difficulty in summarizing an article?</p> <p>How do you summarize an article?</p> <p>How do you feel about using a summary tool?</p>

	What do you wish to accomplish when using a summary tool
--	--

3.7 FRAMEWORK OF THE SYSTEM

3.7.1 High-level system framework for previous text summarization

The system at a high level of abstraction basically allows the visitors to the site to view a summary of previously uploaded documents by signed-up users. This implements text summarization to automatically create an abstract of the uploaded document which the visitors to the website can read. Interested visitors may want to get the full document. This is only possible for signed up users hence the system compels such visitor to sign up and hence has the privileges of other signed up users such as ability to download full document and ability to upload your own documents. A handy tool for signed up users is also an ability to automatically create your document abstract using the summarization engine embedded within the system.

3.7.2 Proposed framework for article summarization

Below is the architecture for the summarization engine already discussed in the chapter two of this work;

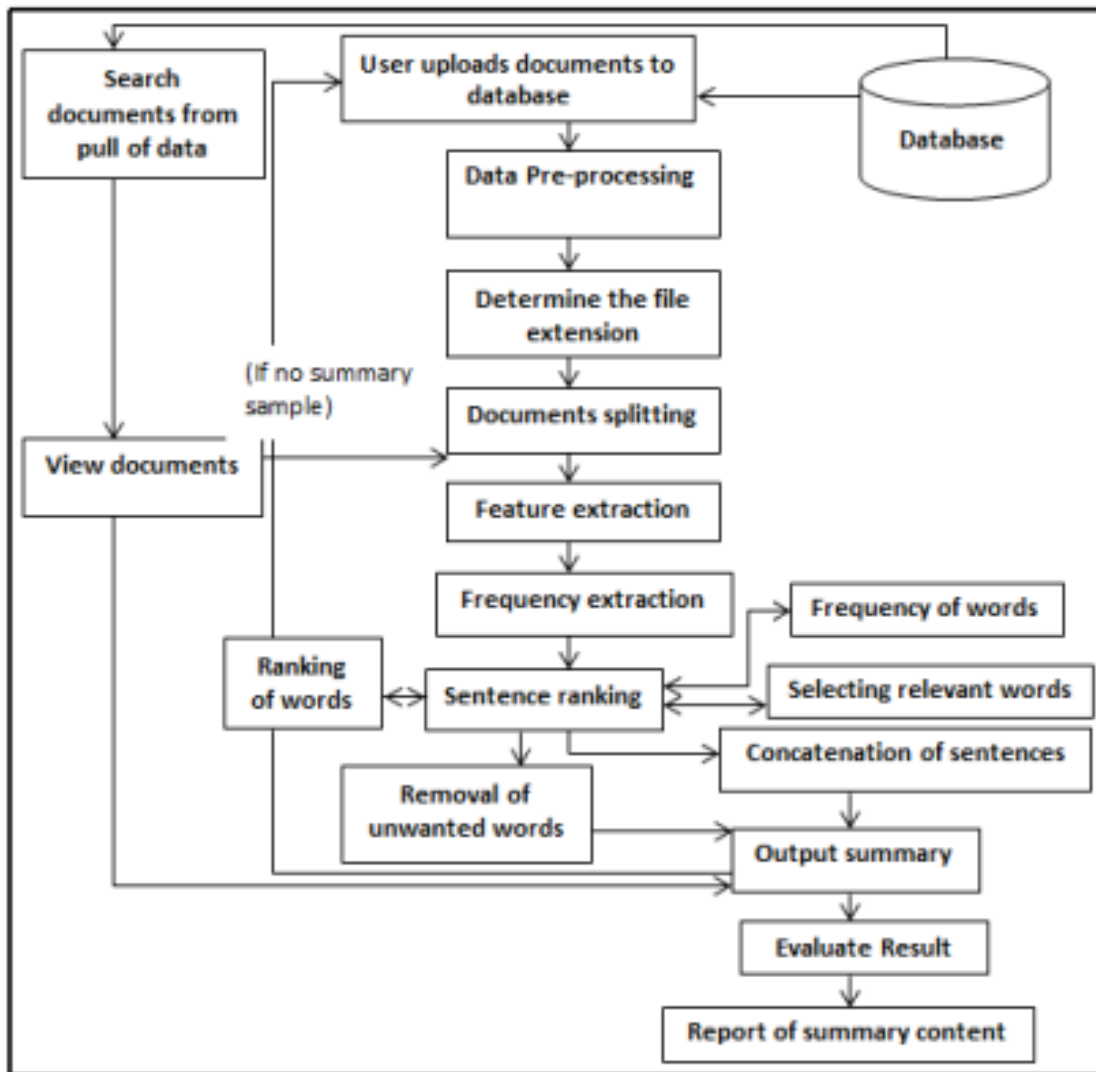


Fig 5: Adapted from Source: (Rakesh et al, 2007)

3.7 CONCLUSION

In conclusion, methodology is the systematic, theoretical analysis of the methods applied in developing a software. Agile methodology has been chosen as the adopted methodology for this project because of the nature of the project and the advantages attached to Agile methodology with respect to the project. Various information gathering and system analysis and design techniques are considered so as to come up with an accurate system.

CHAPTER FOUR

SYSTEM ANALYSIS AND DESIGN

4.1 INTRODUCTION

Requirements analysis is the process of determining user expectations for a new or changed product. These qualities, referred to as criteria, must be quantitative, relevant, and comprehensive. In software engineering, such requirements are commonly referred to as functional specifications. The study of needs is a critical component of any project. Frequent communication with system users to determine specific feature expectations, resolving conflict or ambiguity in requirements as demanded by various users or groups of users, avoiding feature creep, and documenting all aspects of the project development process from beginning to end are part of requirements analysis. Knowledge of hardware, software, and human factors engineering, as well as interpersonal skills, are required for requirements analysis.

4.2 SYSTEM DESIGN

The process of defining the elements of a system such as the architecture, modules, and components, the various interfaces of those components, and the data that flows through that system is known as system design. It is intended to meet specific needs and requirements of a business or organization by engineering a coherent and well-functioning system. A systematic approach to system design is implied by the term systems design. It may take a bottom-up or top-down approach, but in either case, the process is systematic in that it considers all related variables of the system that needs to be created—from the architecture to the necessary hardware and software, all the way down to the data and how it travels and transforms throughout its journey through the system. Then there's systems analysis, systems engineering, and systems architecture to consider. When engineers were attempting to solve complex control and communications problems just before World War II, the systems design approach first appeared. They needed to be able to formalize their work into a formal discipline with proper methods, particularly in new fields such as information theory, operations research and computer science generally.

4.2.1 Feasibility Study

A feasibility study is simply an evaluation of the viability of a proposed project plan or method. This is accomplished through an examination of technical, economic, legal,

operational, and time feasibility factors. Feasibility studies have several advantages, including assisting project owners in determining the pros and cons of undertaking a project before investing significant time and capital into it. Feasibility studies can also provide critical information to a project owner preventing them from entering into a risky and unattainable venture.

Feasibility analysis Facts;

- Technical feasibility
- Economic feasibility
- Operational feasibility
- Schedule feasibility
- Resource feasibility

4.2.1.1 Technical Feasibility

By evaluating the new system's technological viability, one may determine whether it will function as intended and whether the proposed system can be built. The technical evaluation provides information on issues including whether the system's required technology is available, how challenging it will be to construct, and if the knowledge and expertise now held are sufficient to operate it.

The following are some of the technical concerns with feasibility that were brought up during the feasibility stage of the investigation:

- The software (Visual Studio Code) is running on a windows operating system
- The minimum hardware required is 2.4 GHz Intel Core 2.
- The system can be easily expanded

4.2.1.2 Economic Feasibility

Finding out what positive economic advantages the proposed system would provide to the company is the goal of the economic feasibility evaluation. All of the anticipated advantages are identified and quantified. A cost-benefit analysis often forms part of this evaluation. Economic evaluation, which deals with elements that can be measured, quantified, and compared in monetary terms, is an essential component of investment appraisal. This feasibility analysis compares the development and operating costs to show the prefect's tangible and intangible advantages.

4.2.1.3 Operational Feasibility

Operational feasibility is a metric used to assess how successfully a proposed system addresses issues, seizes opportunities, and complies with requirements found during the requirements analysis stage of system development. In terms of development timetable, delivery date, corporate culture, and current business procedures, the operational feasibility study examines how well the planned development projects fit into the existing business environment and objectives.

4.2.1.4 Schedule Feasibility

The system is socially feasible because it can be delivered within the required deadline given our technical expertise.

4.2.1.5 Resource Feasibility

This raises issues like how much time is available to develop the new system, when it can be developed, if it obstructs regular company activities, the kind and quantity of resources needed, dependencies, etc. These plans also include mitigation and contingency measures, ensuring that the business is prepared in case the project does go over budget.

4.3 DESIGN AND DEVELOPMENT PROCESS OF THE SYSTEM

4.3.1 Understanding the problem

A problem statement is a brief description of the problem this project is trying to address. A problem statement identifies the current state, the desired future state, and the gap between the two. A problem statement is an important communication tool that helps ensure that everyone involved in a project understands the issues to be addressed and why this project is important. And the problem definition has been stated clearly in chapter 1 of this research paper.

4.3.2 Design and Development

In this Phase, the SRS document created is transformed into a more logical structure so that it can later be implemented in a programming language. Operations, training, and maintenance plans are all created so that when developing we know what to do at each subsequent stage of the cycle. The development phase is where the actual codes are written and the application is built according to previous design documents and outlined specifications.

Different programming languages are usually used to develop a system for this system we will be using python to achieve the machine learning part and HTML, CSS and JavaScript to achieve the frontend.

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <meta charset="utf-8">
5     <title>Zina summary</title>
6     <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@4.3.1/dist/css/bootstrap.min.css" integrity="sha384-g
7     <link rel="stylesheet" href="/staticFiles/home.css">
8   </head>
9   <body>
10     <div>
11       <nav class="navbar navbar-light bg-light">
12         <a class="navbar-brand" href="#">Zina summary</a>
13       </nav>
14     </div>
15
16     <div class="container">
17       <div class="row">
18         <div class="col-sm mr-1">
19           <div id="colHeader">
20             <h4>Article</h4>
21           </div>
22           <form id="form" action="{{ url_for('submit') }}" method="post" enctype = "multipart/form-data">
23             <textarea id="article" name="article">
24               placeholder="Copy and past your text or url (Only one url per request)" >{{ articleText }}</textarea>
25             <input type = "file" name = "file" />
26             <input id="btn" type="submit" value="Summarize" />
27           </form>
28         </div>
29         <div class="col-sm">
30           <div id="colHeader">
31             <h4>Summary</h4>
32           </div>
33           <p>
34             {{ summary }}
35           </p>
36         </div>
37       </div>
38     </div>
39
```

Fig2. HTML code

```
--danger: #dc3545;
--light: #f8f9fa;
--dark: #343a40;
--breakpoint-xs: 0;
--breakpoint-sm: 576px;
--breakpoint-md: 768px;
--breakpoint-lg: 992px;
--breakpoint-xl: 1200px;
--font-family-sans-serif: -apple-system,BlinkMacSystemFont,"Segoe UI",Roboto,"Helvetica Neue",Arial,"Noto Sans",
  serif,"Apple Color Emoji","Segoe UI Emoji","Segoe UI Symbol","Noto Color Emoji";
--font-family-monospace: SFMono-Regular,Menlo,Monaco,Consolas,"Liberation Mono","Courier New",monospace;
}

html {
  font-family: sans-serif;
  line-height: 1.15;
  -webkit-text-size-adjust: 100%;
  -webkit-tap-highlight-color: transparent;
}

*, ::after, ::before {
  box-sizing: border-box;
}

html {
  display: block;
}

Pseudo ::before element

*, ::after, ::before {
  box-sizing: border-box;
}

Pseudo ::after element

*, ::after, ::before {
  box-sizing: border-box;
}
```

Fig 3 CSS

```

1  from flask import Flask, request, render_template
2  from werkzeug import secure_filename
3
4  import os
5  import requests
6  import validators
7
8  app = Flask(__name__, static_folder='staticFiles')
9
10 ALLOWED_EXTENSIONS = ['txt', 'pdf']
11
12 def getSummary(article):
13     # Define the API endpoint and API key
14     base_url = "https://api.smmry.com"
15     API_KEY = "330B752F19"
16     response = None
17     params = {"SM_API_KEY": API_KEY}
18
19     if validators.url(article):
20         params["SM_URL"] = article
21         # Make the API request
22         response = requests.get(base_url, params=params)
23     else:
24         data = {"sm_api_input": article}
25         header_params = {"Expect": "100-continue"}
26         # Make the API request
27         response = requests.post(
28             base_url, params=params, data=data, headers=header_params)
29
30     # Get the summary from the response
31     summary = response.json()["sm_api_content"]
32
33     return summary
34
35
36 def allowed_file(filename):
37     return '.' in filename and \
38         filename.rsplit('.', 1)[1].lower() in ALLOWED_EXTENSIONS
39

```

Fig 4 Python code

4.3.3 Evaluation

Evaluation includes performing functional testing such as unit testing, code quality testing, integration testing, system testing, security testing, performance testing, acceptance testing, and non-functional testing. If an error is detected, the developer will be notified. A verified (actual) error is fixed and a new version of the software is created. The best way to ensure that all tests run regularly is to implement automated tests. Continuous integration tools support this need.

4.3.4 Implementation and Integration

After testing, the overall design of the software is put together. Various modules or designs are integrated into the main source code through the efforts of developers. A training environment is typically used to detect further errors and defects. Information systems are integrated into the environment and eventually

installed. After this phase, the software is theoretically ready for the market and available to all end users.

4.3.5 Maintenance

In the maintenance mode we practice the necessary activities to resolve issues reported by end users. Additionally, the developer is responsible for implementing any necessary changes to the software after deployment. This may include addressing remaining bugs that could not be patched prior to release or resolving new issues that surfaced based on user reports. Larger systems may require longer maintenance windows than smaller systems.

4.4 MACHINE LEARNING

The brain of machine learning is where all learning occurs. The way a computer learns is comparable to how a person learns. Experience is how people learn. The easier it is to forecast, the more we know. By analogy, our chances of success are lower than they would be in a known circumstance when we encounter one. Machines receive the same training. The computer observes an example in order to create a precise forecast. The machine is capable of predicting the result when we provide a comparable case. However, much like a person, the machine has trouble predicting if it is given a new example. Learning and inference are at the heart of machine learning. The first way the machine learns is by identifying patterns.

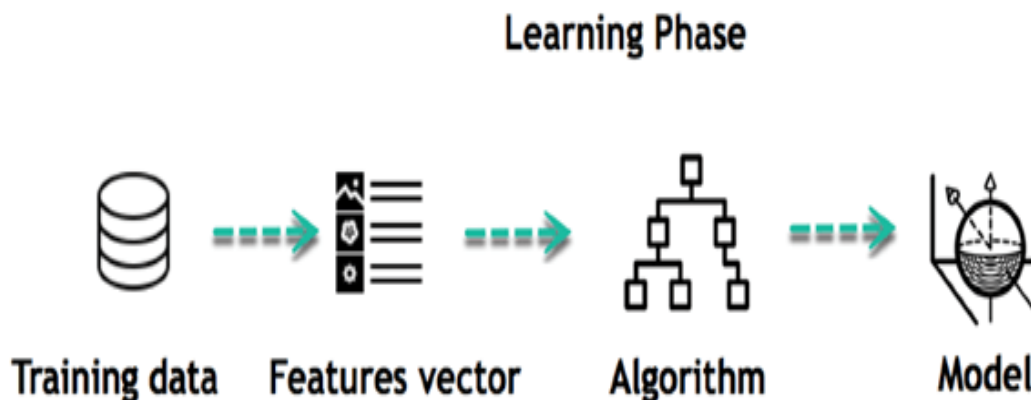


Figure 8. Machine learning phase

4.5 SYSTEM MODELLING

4.5.1 Use Case Diagram

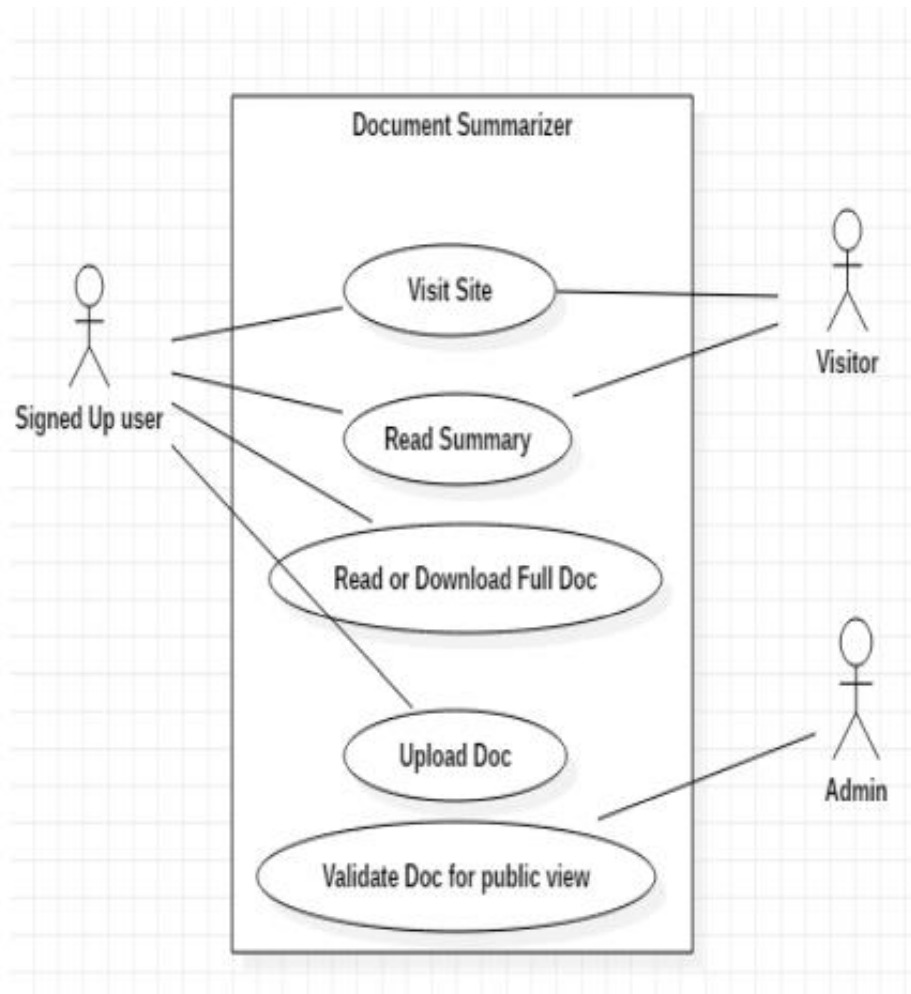


Figure 9 use case diagram for automated article summarization

4.6 System Implementation

4.6.1 System Testing

The process of testing demonstrates that a program complies with its specification and operates as intended. Testing is used to find mistakes or circumstances where the program's behavior is unacceptable. Testing aims to demonstrate thoroughness, enhance software quality, and offer maintenance assistance. Several guidelines that can work well as testing objectives;

- Testing is the process of executing a program with intent of finding errors.

- A good test case is one that has a high probability of finding an as-yet-undiscovered error.
- A successful test is one that uncovers an as-yet-undiscovered error.

The suggested system was tested using the following methodologies:

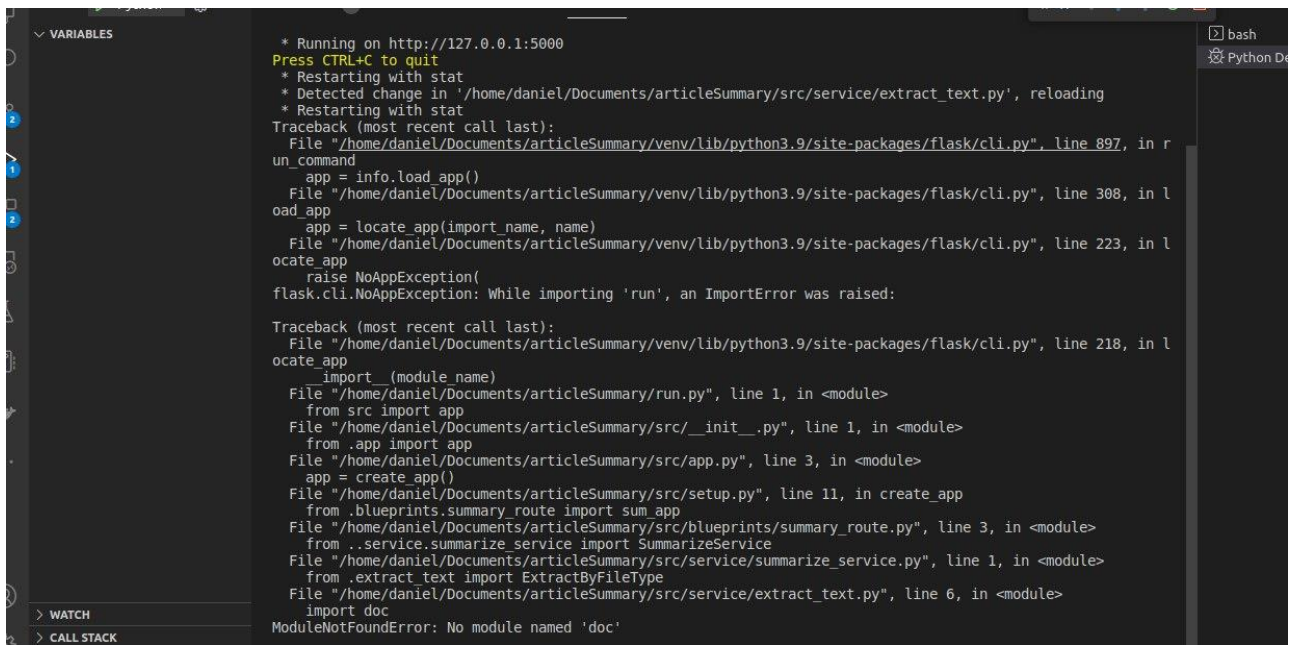
Careful scrutiny of procedural detail is the cornerstone of white-box software testing. By providing test cases that put certain sets of circumstances and/or loops to the test, the program is tested along its logical routes. It is possible to check the "state of the program" at various times to see if the asserted or expected status matches the actual situation.

The focus of black-box testing, also known as behavioral testing, is on the functional specifications of software. The software engineer may determine the input circumstances that will completely exercise all of a program's requirements using this testing methodology. Black box testing attempts to find the errors like;

- Incorrect functions
- Interface errors
- Mistakes while accessing other databases or data structures
- Mistakes in behavior or performance
- Initiation and termination mistakes

In Black box testing software is exercised over a full range of inputs and outputs are observed for correctness.

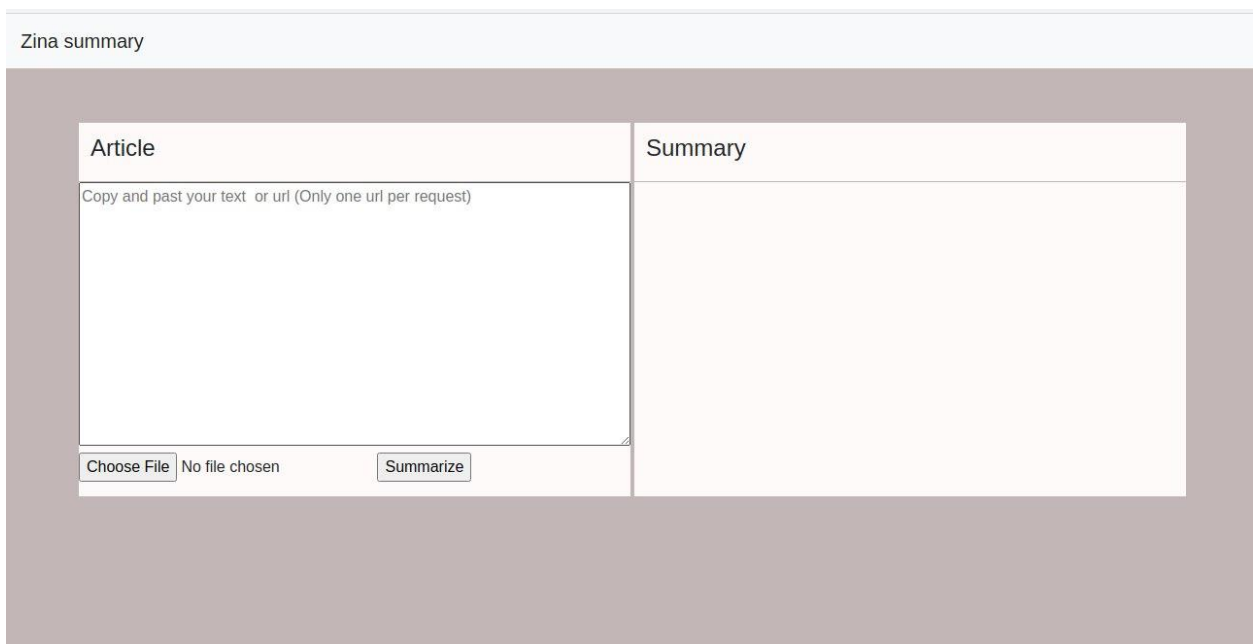
Testing images for Z-Summary



```
* Running on http://127.0.0.1:5000
Press CTRL+C to quit
* Restarting with stat
* Detected change in '/home/daniel/Documents/articleSummary/src/service/extract_text.py', reloading
* Restarting with stat
Traceback (most recent call last):
  File "/home/daniel/Documents/articleSummary/venv/lib/python3.9/site-packages/flask/cli.py", line 897, in run_command
    app = info.load_app()
  File "/home/daniel/Documents/articleSummary/venv/lib/python3.9/site-packages/flask/cli.py", line 308, in load_app
    app = locate_app(import_name, name)
  File "/home/daniel/Documents/articleSummary/venv/lib/python3.9/site-packages/flask/cli.py", line 223, in locate_app
    raise NoAppException(
flask.cli.NoAppException: While importing 'run', an ImportError was raised:

Traceback (most recent call last):
  File "/home/daniel/Documents/articleSummary/venv/lib/python3.9/site-packages/flask/cli.py", line 218, in locate_app
    import_(module_name)
  File "/home/daniel/Documents/articleSummary/run.py", line 1, in <module>
    from src import app
  File "/home/daniel/Documents/articleSummary/src/__init__.py", line 1, in <module>
    from .app import app
  File "/home/daniel/Documents/articleSummary/src/app.py", line 3, in <module>
    app = create_app()
  File "/home/daniel/Documents/articleSummary/src/setup.py", line 11, in create_app
    from .blueprints.summary_route import sum_app
  File "/home/daniel/Documents/articleSummary/src/blueprints/summary_route.py", line 3, in <module>
    from ..service.summarize_service import SummarizeService
  File "/home/daniel/Documents/articleSummary/src/service/summarize_service.py", line 1, in <module>
    from .extract_text import ExtractByFileType
  File "/home/daniel/Documents/articleSummary/src/service/extract_text.py", line 6, in <module>
    import doc
ModuleNotFoundError: No module named 'doc'
```

Fig 5. File search error



Zina summary

Article	Summary
<p>Copy and past your text or url (Only one url per request)</p> <p><input type="button" value="Choose File"/> No file chosen <input type="button" value="Summarize"/></p>	

Fig 6. Input Testing Phase

Not Found

The requested URL was not found on the server. If you entered the URL manually please check your spelling and try again.

Fig 7. Error output for wrong input

```
Feb 21 09:37:42 PM ==> Cloning from https://github.com/Dani212/articleSummary...
Feb 21 09:37:43 PM ==> Checking out commit 20441b6d35ec71b27baee619d0096f6b95cc216d in branch main
Feb 21 09:37:48 PM ==> Using Python version: 3.7.10
Feb 21 09:37:51 PM ==> Running build command './build.sh'...
Feb 21 09:37:52 PM Collecting pip
Feb 21 09:37:52 PM   Downloading pip-23.0.1-py3-none-any.whl (2.1 MB)
Feb 21 09:37:52 PM Installing collected packages: pip
Feb 21 09:37:52 PM   Attempting uninstall: pip
Feb 21 09:37:52 PM     Found existing installation: pip 20.1.1
Feb 21 09:37:53 PM     Uninstalling pip-20.1.1:
Feb 21 09:37:53 PM       Successfully uninstalled pip-20.1.1
Feb 21 09:37:54 PM Successfully installed pip-23.0.1
Feb 21 09:37:56 PM Processing /home/ktietz/src/ci/alabaster_1611921544520/work
Feb 21 09:37:56 PM ERROR: Could not install packages due to an OSError: [Errno 2] No such file or directory: '/home/ktietz/src/ci/alabaster_1611921544520/work'
Feb 21 09:37:56 PM
Feb 21 09:37:56 PM ==> Build failed 🙄
Feb 21 09:37:56 PM ==> Generating container image from build. This may take a few minutes...
```

Scroll to bottom

Fig 8 system build error

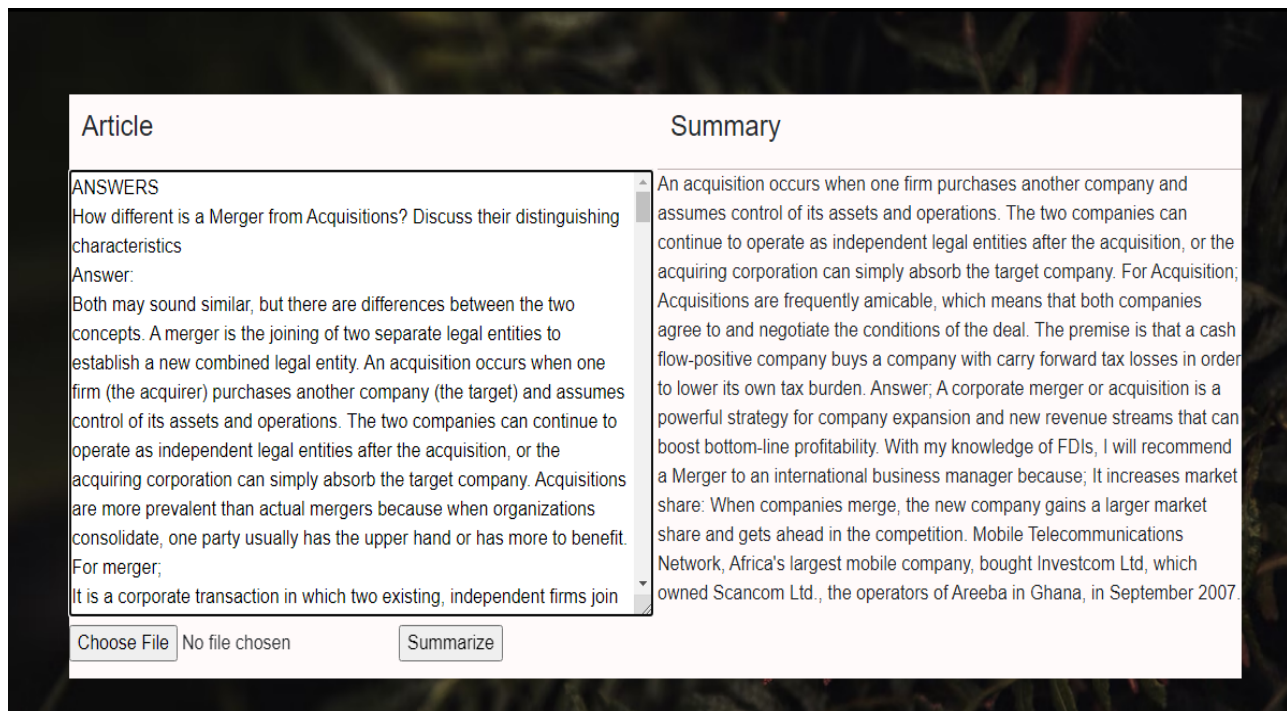


Fig 9. Summary Output

4.6.2 Software Testing Strategies

The strategies for testing the system involve;

- Unit Testing: Unit testing is the basic level of testing. Unit testing is used to make sure that every software has been thoroughly tested.
- Integration testing: Integration testing is the next phase. To make sure the software requirements are satisfied, separate program components or programs are combined and tested as a whole system.
- Acceptance testing: Planning and carrying out various sorts of tests as part of acceptance testing is necessary to show that the developed software system fits the requirements. Finally, after passing all of the testing phases, our product satisfies the criteria.

4.7 CONCLUSION

The creation and execution of An Automatic text system were discussed in this chapter. This system was created using the Agile technique. The UML methodology was used to evaluate

the design and functions of the system. The system functionalities were designed using a use case diagram. This chapter also covered the system testing technique.

CHAPTER FIVE

FINDINGS, CONCLUSION AND RECOMMENDATIONS

5.1 INTRODUCTION

With the growth of the internet and big data, individuals are becoming overwhelmed by the vast amount of information and documents available on the internet. Many researchers are motivated to develop a technology solution that can automatically summarize texts as a result of this. Automatic text summarizing provides summaries that incorporate all relevant information from the original material and include important sentences (Allahyari et al., 2017; Gambhir and Gupta, 2017). As a result, the information is delivered swiftly while maintaining the document's original objective (Murad and Martin, 2007). Text summarization research has been explored since the mid-twentieth century, when Lun (1958) first described it openly using a statistical technique called word frequency diagrams. To date, numerous techniques have been developed. There are single and multi-document summary options based on the number of documents. Meanwhile, the extractive and abstractive outcomes are based on the summary results.

Researchers are increasingly using automatic text summarizing (ATS). The method of creating a subset of the main text is known as automatic text summarization. This subset of the main text represents the entire text as well as the text's major theme. Text summarization is another name for automatic text summarizing. ATS is a vital branch of Natural Language Processing (NLP) and Data Mining (DM). This comprises the text's abstractive and extractive summaries.

The construction phase of the project began once the thorough design for individual modules was finished. During this phase, integration of the numerous source codes into the overall system was accomplished. for system users to quickly become accustomed to the system. This chapter's main goal is to describe the transition from the system design phase to the operational system and overall, this chapter also includes an overview.

Automatic Text Summary implements web-based technologies such as;

- Python
- Visual Studio Code

The system's implementation tools must take the project's requirements and scope into account. The developer's experience must also be taken into account. However, the developer's insufficient experience might have an impact on the creation of a solution.

5.2 FUTURE WORKS AND RECOMMENDATIONS

The researcher advises moving on with the system's implementation while evaluating it for the addition of new modules or components that have been left out due to time and financial constraints.

Users using this system should;

- As new issues occur, increase the number of solutions.
- Regularly copy the summary text in case of a calamity or system breakdown.
- Make sure that hardware components are constantly functioning properly. If necessary, replacement should be made.

The developer should be engaged for a system or program upgrade as the system needs grow.

5.3 SUMMARY

We are in an era of technological advancement and rapid progress in almost all aspects of living. It is therefore of great importance if our articles, texts and documents are computerized to meet the increasing needs in summary. The project's objective was to find a more efficient and easy way of carrying out the functions of helping people summarize their articles or documents including students to save difficulties encountered during reading. In achieving the general objective mentioned, this project will specifically concentrate on the following:

- The system shall allow users to access the webpage.
- The system shall enable users upload documents online.
- The system shall provide output to users.
- The system shall help in generating an easy text summary for every file, texts or URL
- The system shall maintain better user relationship.

This project gives an overview of previous research and studies in the topic of Automatic Text Summarization.

5.4 CONCLUSION

In the chapter one, the problem statement of current system was explained and also the project objectives which are to design An Automatic Text Summary using Text Rank. In chapter two the researcher conducted a literature review of existing systems. In Chapter three, the researcher underscored the various methodologies that will be used to implement the system. Chapter four, the researcher used the methodology stated in chapter three to build the system thus, System Analysis, System Design, System implementation and testing the developed system.

REFERENCES:

1. SaiyedSaziyabegum, Priti S. Sajja, "Literature Review on Extractive Text Summarization Approaches" International Journal of Computer Applications (0975 - 8887) Volume 156-No 12, December 2016
2. Reeta Rani, Sawal Tandon, "LITERATURE REVIEW ON AUTOMATIC TEXT SUMMARIZATION" International Journal of Current Advanced Research (9779 - 9783) Volume 7; Issue 2(C); February 2018
3. Chin-Yew Lin, "Rouge: A Package for Automatic Evaluation of Summaries." Barcelona Spain, Workshop Text Summarization Branches Out, Post- Conference Workshop of ACL 2004
4. Akshi Kumar, Aditi Sharma, Sidhant Sharma, Shashwat Kashyap, "Performance Analysis of Keyword Extraction Algorithms Assessing Extractive Text Summarization." International Conference on Computer, Communication, and Electronics (Comptelix), 2017.
5. Pankaj Gupta, Ritu Tiwari and Nirmal Robert, "Sentiment Analysis and Text Summarization of Online Reviews: A Survey." International Conference on Communication and Signal Processing, 2016.
6. Taeho Jo, "K Nearest Neighbor for Text Summarization using Feature Similarity." International Conference on Communication, Control, Computing and Electronics Engineering (ICCCCEE), 2017
7. S. Hima Bindu Sri, Sushma Rani Dutta, "A Survey on Automatic Text Summarization Techniques" International Conference on Physics and Energy 2021 (ICPAE 2021)
8. Donalek, Ciro. "Supervised and unsupervised learning." Astronomy Colloquia. USA. 2011.
9. Jawahar, D. (2012). Overview of System Analysis & Design. Prof. Dharminder Kumar
10. Barros, C., Lloret, E., Saquete, E., Navarro-colorado, B., 2019. NATSUM: Narrative abstractive summarization through cross- document timeline generation. Information Process. Manag. 56, 1775–1793.
11. PAULIINA ANTILA, "Automatic Text Summarization Master's Thesis" Computer Science 51 p May 2018.

12. Allahyari, M., Pouriye, S., Assefi, M., Safaei, S., Trippe, E.D., Gutierrez, J.B., Kochut, K., 2017. Text summarization techniques: A brief survey. *Int. J. Adv. Comput. Sci. Appl.* 8.
13. Murad, M.A.A., Martin, T., 2007. Similarity-based estimation for document summarization using fuzzy sets. *Int. J. Comput. Sci. Secur.* 1, 1.
14. Lun, H.P., 1958. The automatic creation of literature abstracts. *IBM J.*, 159–165
15. Adhika Pramita Widyassari, Supriadi Rustad, Guruh Fajar Shidik, Edi Noersasongko,
16. Abdul Syukur, Affandy Affandy, De Rosal Ignatius Moses Setiadi, "Review of automatic text summarization techniques & methods" A.P. Widyassari et al. / *Journal of King Saud University – Computer and Information Sciences* 34 (2022) 1029–1046.
17. Olivier Chapelle, Bernhard Schölkopf, Alexander Zien, and others. (2006). *Semisupervised learning*. Vol. 2. MIT press Cambridge.
18. Ping Chen and Rakesh Verma. (2006). A query-based medical information summarization system using ontology knowledge. In *Computer-Based Medical Systems, 2006. CBMS 2006. 19th IEEE International Symposium on*. IEEE, 37– 42.
19. Freddy Chong Tat Chua and Sitaram Asur. (2013). Automatic Summarization of Events from Social Media.. In *ICWSM*.
20. John M Conroy and Dianne P O' leary. (2001). Text summarization via hidden Markov models. In *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, 406– 407.
21. Hal Daumé III and Daniel Marcu. (2006). Bayesian query-focused summarization. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, 305– 312.
22. Scott C. Deerwester, Susan T Dumais, Thomas K. Landauer, George W. Furnas, and Richard A. Harshman. (1990). Indexing by latent semantic analysis. *JASIS* 416 (1990), 391– 407.
23. J-Y Delort, Bernadette Bouchon-Meunier, and Maria Rifqi. (2003). Enhanced web document summarization using hyperlinks. In *Proceedings of the fourteenth ACM conference on Hypertext and hypermedia*. ACM, 208-215.

24. Ted Dunning. (1993). Accurate methods for the statistics of surprise and coincidence. Computational linguistics 19, 1 (1993), 61-74.
25. Harold P Edmundson. (1969). New methods in automatic extracting. Journal of the ACM (JACM) 16, 2 (1969), 264-285.
26. Günes Erkan and Dragomir R Radev. (2004). LexRank: Graph-based lexical centrality as salience in text summarization. J. Artif. Intell. Res.(JAIR) 22, 1 (2004), 457– 479.
27. Yihong Gong and Xin Liu. (2001). Generic text summarization using relevance measure and latent semantic analysis. In Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval. ACM, 19-25.
28. Vishal Gupta and Gurpreet Singh Lehal. (2010). A survey of text summarization extractive techniques. Journal of Emerging Technologies in Web Intelligence 2, 3 (2010), 258-268.
29. Ben Hachey, Gabriel Murray, and David Reitter. (2006). Dimensionality reduction aids term co-occurrence based multi-document summarization. In Proceedings of the workshop on task-focused summarization and question answering. Association for Computational Linguistics, 1 – 7.

APPENDIX

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <meta charset="utf-8">
5     <title>Zina summary</title>
6     <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@4.3.1/dist/css/bootstrap.min.css" integrity="sha384-ga2wVarn0D13PpGRsfhh6lc/1jP1bE1eeCt0J365+ecfdgdy" crossorigin="anonymous">
7     <link rel="stylesheet" href="/staticFiles/home.css">
8   </head>
9   <body>
10     <div>
11       <nav class="navbar navbar-light bg-light">
12         <a class="navbar-brand" href="#">Zina summary</a>
13       </nav>
14     </div>
15
16     <div class="container">
17
18       <div class="row">
19         <div class="col-sm mr-1">
20           <div id="colHeader">
21             <h4>Article</h4>
22           </div>
23           <form id="form" action="{{ url_for('submit') }}" method="post" enctype = "multipart/form-data">
24             <textarea id="article" name="article"
25               placeholder="Copy and past your text or url (Only one url per request)" >{{ articleText }}</textarea>
26             <input type = "file" name = "file" />
27             <input id="btn" type="submit" value="Summarize" />
28           </form>
29         </div>
30         <div class="col-sm">
31           <div id="colHeader">
32             <h4>Summary</h4>
33           </div>
34           <p>
35             {{ summary }}
36           </p>
37         </div>
38       </div>
39     </div>
```

VSCODE 1 – html/ bootstrap CSS

Article	Summary
<p>A survey is a simple tool for gathering information. A poll usually consists of a series of questions designed to assess a participant's preferences, attitudes, traits, and opinions about a particular topic. As a research method, questionnaires allow concepts to be counted or quantified.</p> <p>That is, use a wider sample or subset of the audience, which allows the findings to be applied to a wider population. For example, in a given year he has 100,000 unique users of his website. If he collects information from 2,000 of these users, he can confidently apply</p> <p>Summarize</p>	<p>There are different types of questions in surveys, depending on the circumstances of the survey and the type of information you want to access. Many surveys combine open-ended and closed-ended questions, such as rating scales and semantic scales. Paper surveys are a more traditional method of data collection and can easily lose data. Surveys can be effective for identifying: Who your users are; What your users want; What they purchase; Where they shop; What they own Benefits of Surveys are: It provides information to better understand end-users in order to develop better products. Disadvantages of Surveys are: The validity of the research data can be affected by survey response bias. High survey dropout rates can also affect the number of responses received in your survey. Structured interviews are similar to surveys, with the same questions asked in the same order for each topic and multiple-choice answers.</p>

Interface I – output display for copy and paste / URL

```

app.py > ...
1  from flask import Flask, request, render_template
2  from werkzeug import secure_filename
3
4  import os
5  import requests
6  import validators
7
8  app = Flask(__name__, static_folder='staticFiles')
9
10 ALLOWED_EXTENSIONS = ['txt', 'pdf']
11
12 def getSummary(article):
13     # Define the API endpoint and API key
14     base_url = "https://api.smmry.com"
15     API_KEY = "33DB752F19"
16     response = None
17     params = {"SM_API_KEY": API_KEY}
18
19     if validators.url(article):
20         params["SM_URL"] = article
21         # Make the API request
22         response = requests.get(base_url, params=params)
23     else:
24         data = {"sm_api_input": article}
25         header_params = {"Expect": "100-continue"}
26         # Make the API request
27         response = requests.post(
28             base_url, params=params, data=data, headers=header_params)
29
30     # Get the summary from the response
31     summary = response.json()["sm_api_content"]
32
33     return summary
34
35
36 def allowed_file(filename):
37     return '.' in filename and \
38         filename.rsplit('.', 1)[1].lower() in ALLOWED_EXTENSIONS
39

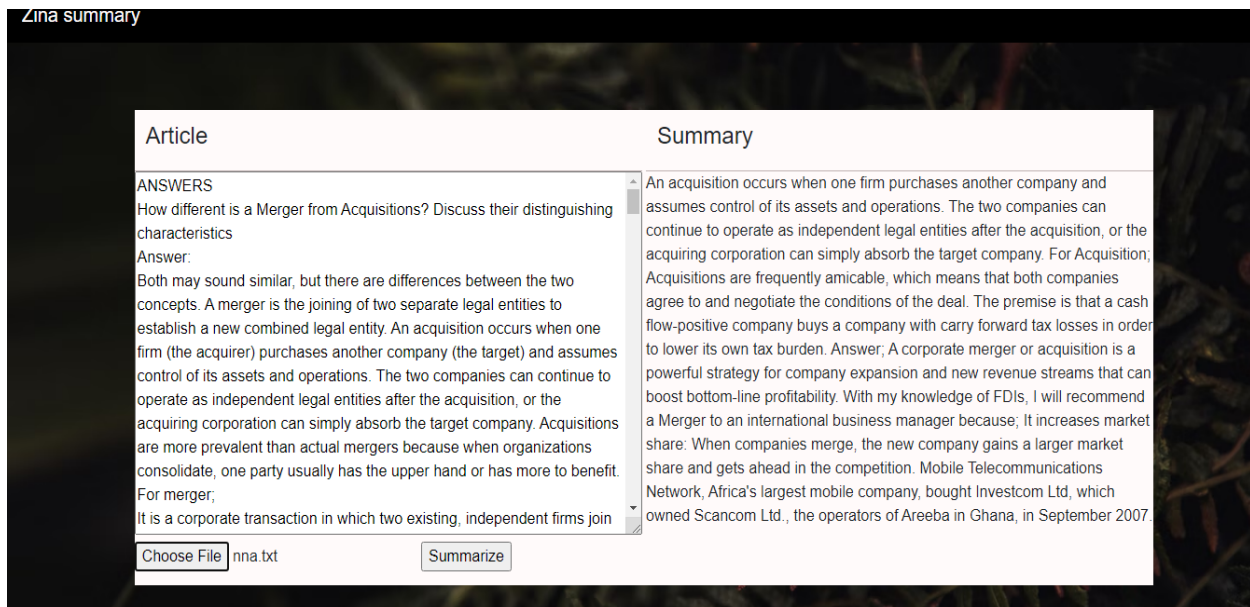
```

VSCODE II - Python

Zina summary

Article	Summary
<p>Copy and past your text or url (Only one url per request)</p> <div> <input type="button" value="Choose File"/> No file chosen <input type="button" value="Summarize"/> </div>	

Interface II – pdf / txt file upload



Interface 3 – summary output for pdf / txt file upload

```

39 |
40 |
41 | @app.route("/")
42 | def hello_world():
43 |     return render_template('home.html')
44 |
45 |
46 | @app.post('/submit')
47 | def submit():
48 |     articleText = ""
49 |     file
50 |     if 'file' not in request.files:
51 |         articleText = request.form['articlee']
52 |         file = request.files['file']
53 |
54 |     if file and allowed_file(file.filename):
55 |         filename = secure_filename(file.filename)
56 |         file.save(os.path.join(app.config['UPLOAD_FOLDER'], filename))
57 |         articleText = file.read()
58 |
59 |     summary = getSummary(articleText)
60 |     return render_template('home.html', summary=summary, articleText=articleText)
61 |
62 |
63 | if __name__ == '__main__':
64 |     app.run(host="0.0.0.0", port=5000, debug=True)
65 |

```

VSCODE III - Python