

INFORMATICS INSTITUTE OF TECHNOLOGY

In Collaboration with
UNIVERSITY OF WESTMINSTER



SinDOC:

**A Combined Approach of Summarizing Low Resource Sinhala
Language Documents**

A dissertation by
Mr. Mohamed Hamza Ziyad

Supervised by
Mr. Mithushan Jalangan

Submitted in partial fulfillment of the requirements for the BSc. (Hons) Computer Science
degree at the University of Westminster.

May 2023

DECLARATION

I hereby declare that this dissertation and all resources related to it are my own work. Any of the contents that have been stated in here are not submitted or shared across any other platforms or courses. All details extracted from other sources are cited within the research.

Name – Mohamed Hamza Ziyad

Registration No. – w1761340/2019407

Date: 10/05/2022

Signature:

A handwritten signature in black ink, appearing to read 'Mohamed Hamza Ziyad', written over a light blue grid background.

ABSTRACT

Summarization is one of the NLP related tasks that is widely researched during the recent times. Due to the large amount of data stored on the internet, users do not consume all information that is available. They will be looking for a small piece of information that will convey the most important information to them as quickly as possible. Sinhala language is one of the low resource languages which do not have many contributions within the field of document summarization. Mainly because of the lack of resources. To address this problem the author has come up with a novel approach that uses a combined method to summarize Sinhala documents by using both extractive and abstractive techniques.

The proposed model uses word frequency and sentence scoring approaches for the extractive model and uses a pre-trained model for the abstractive summarization model. The author has presented a new dataset that has been translated from an existing English dataset. The pre-trained model uses this dataset for training. The author was also able to prove that automating hyper parameter tuning to generate training arguments for the abstractive model gives better results with less time constraints when compared to the traditional approaches. The author has also given the option of generating summaries through all three approaches to make sure that the user gets the best overall result. The model build in this research shows good results that outperforms previous works.

Overall, this research proposes a combined approach of generating summaries for Sinhala documents. As all the models and code are made publicly available the author believes that this work could build a strong foundation for future researchers.

Keywords: Natural Language Processing, Extractive summarization, Abstractive summarization, Combined summarization, Sinhala Language

Subject Descriptors:

- Computing methodologies → Artificial Intelligence → Natural language processing → Information extraction
- Theory of computation → Theory and algorithms for application domains → Machine learning theory → Semi-supervised learning
- Human centered computing → Interaction design → Interaction design process and methods → User centered design

ACKNOWLEDGEMENT

I am so grateful for all those who have been a great part of my journey by helping me successfully complete this dissertation. Without any of their support it would never been possible for me to complete this project.

Firstly, I would like to thank my mentor Mr. Mithushan Jalangan who has been always supportive. Your support during hard times has helped me a lot on motivating myself to keep going with the project without losing hope. I am deeply grateful to have you as my mentor as you were always ready to provide guidance and being kind enough to attend to any issue.

I would then like to thank my lectures who have been a great support by providing all necessary materials and guidance needed to complete this research project. I would like to specially thank Mr. Guhanathan Poravi for been a quality lecturer. Your explanations on how to work on the final project has immensely helped me throughout the project.

I owe my parents a debt of gratitude. They have always been supportive throughout my academic journey. Your love, encouragement and advice has kept me strong and helped me face and hardships at any time. I am eternally grateful for what you both have done for me.

I would also like to thank my peers for being supportive throughout my research time and kept motivating me on doing better. I am very grateful to have such wonderful friends and mates.

Finally, I would like to thank all the juniors and seniors who were very supportive by helping me solve my problems whenever their assistance was required.

TABLE OF CONTENTS

CHAPTER 01: INTRODUCTION	1
1.1. Chapter Overview	1
1.2. Problem Domain	1
1.2.1. Natural Language Processing (NLP)	1
1.2.2. Low resource language	1
1.2.3. Document summarization	2
1.3. Problem Definition.....	2
1.3.1. Problem Statement	3
1.4. Research Motivation	3
1.5. Research Gap	3
1.6. Contribution To The Body Of Knowledge	4
1.6.1. Contribution to the Problem Domain.....	4
1.7. Research Challenge.....	4
1.8. Research Questions	5
1.9. Research Aim.....	5
1.10. Research Objectives	5
1.11. Chapter Summary	7
CHAPTER 02: LITERATURE REVIEW	8
2.1. Chapter Overview	8
2.2. Concept Map.....	8
2.3. Research Domain	8
2.3.1. Summarization Techniques.....	8
2.3.2. Sinhala Language.....	9
2.3.3. Why Document Summarization for Sinhala Language	10

2.4. Existing Work	11
2.4.1. Traditional Approaches.....	11
2.4.2. Other approaches	15
2.4.3. Summary of existing work.....	16
2.5. Technological Review	19
2.5.1. Proposed Architecture for the Sinhala Document Summarization Application.	19
2.5.2. Algorithmic Approaches	20
2.5.3. Machine Learning Approaches	21
2.5.4. Deep Learning Approaches.....	21
2.5.5. Dataset Preparation	21
2.5.6. Data Pre-processing	22
2.6. Evaluation Methods on Document Summarization	22
2.6.1. Automatic Evaluation	22
2.6.1. Human Evaluation	23
2.6.3. Benchmarking	23
2.9. Chapter Summary	24
CHAPTER 03: METHODOLOGY	25
3.1. Chapter Overview	25
3.2. Research Methodology	25
3.3. Development Methodology	26
3.3.1. Requirement Elicitation Methodology.....	26
3.3.2. Design methodology	26
3.4. Project Management Methodology.....	26
3.4.1. Resource Requirements	27
2.4.3. Schedule.....	29

2.5. Chapter Summary	30
CHAPTER 04: SOFTWARE REQUIREMENT SPECIFICATION	31
4.1. Chapter Overview	31
4.2. Rich Picture Diagram.....	31
4.3. Stakeholder Analysis	32
4.3.1. Stakeholder Onion Model	32
4.3.2. Analysis of the Stakeholder	33
4.4. Selection of Requirement Elicitation Methodology	34
4.5. Discussion of Findings.....	35
4.5.1. Literature Review.....	35
4.5.2. Survey	35
4.5.3. Formal Interview.....	39
4.6. Summary of Findings.....	41
4.7. Context Diagram.....	41
4.8. Use Case Diagram.....	42
4.9. Use Case Description.....	43
4.10. Requirements	44
4.10.1. Functional requirements.....	44
4.10.2. Non-functional requirements	45
4.11. Chapter Summary	46
CHAPTER 05: SOCIAL, LEGAL, ETHICAL AND PROFESSIONAL ISSUES	47
5.1. Chapter Overview	47
5.2. Breakdown of Social, Legal, Ethical and Professional Issues	47
5.3. Chapter Summary	47
CHAPTER 06: DESIGN.....	48

6.1. Chapter Overview	48
6.2. Design Goals	48
6.3. High Level Design	49
6.3.1. Tiered Architecture of the Proposed System	49
6.3.2. Discussion of tiers of the architecture.....	50
6.4. System Design	51
6.4.1. Selection of design paradigm.....	51
6.5. Design Diagrams.....	51
6.5.1. Data flow diagram.....	51
6.5.2. UI design	52
6.5. Chapter Summary	53
CHAPTER 07: IMPLEMENTATION	54
7.1. Chapter Overview	54
7.2. Technology Selection.....	54
7.2.1. Technology Stack.....	54
7.2.2. Dataset Selection.....	54
7.2.3. Development Frameworks	55
7.2.4. Programming Languages	55
7.2.5. Libraries	56
7.2.6. IDE's	56
7.2.7. Summary of Technology Selection.....	57
7.3. Implementing of the core functionality.....	57
7.3.1. Extractive Model Implementation	57
7.3.2. Dataset Creation.....	59
7.3.3. Data pre-processing	61

7.3.4. Abstractive Model Implementation	61
7.3.5. Combine Approach Model Implementation	62
7.5. UI Implementation	63
7.4. Chapter Summary	63
CHAPTER 08: TESTING.....	65
8.1. Chapter Overview	65
8.2. Objectives and Goals of Testing	65
8.3. Testing Criteria	65
8.4. Model Testing	65
8.7.1. sinMT5 Comparisons Before and After Hyper Parameter Tuning	65
8.7.1. Extractive, Abstractive and Combined Summarization Model Results.....	67
8.5. Benchmarking	68
8.6. Functional Testing	68
8.7. Module and Integration Testing.....	70
8.7. Non-Functional Testing	71
8.7.1. Performance	71
8.7.2. Usability	72
8.7.3. Security	72
8.7.4. Maintainability	73
8.8. Limitations of the Testing Process.....	73
8.9. Chapter Summary	73
CHAPTER 09: EVALUATION	74
9.1. Chapter Overview	74
9.2. Evaluation Methodology and Approach	74
9.3. Evaluation Criteria	74

9.4. Self-Evaluation	75
9.5. Selection of the Evaluators	76
9.6. Evaluation Result	77
9.6.1. Technical Experts.....	77
9.6.2. Domain Experts	78
9.6.3. Focus Group.....	79
9.7. Limitation of Evaluation	79
9.8. Evaluation on Functional Requirements	80
9.9. Evaluation on Non-Functional Requirements.....	81
9.10. Chapter Summary	82
CHAPTER 10: CONCLUSION	83
10.1. Chapter Overview	83
10.2. Achievements of Research Aim and Objectives.....	83
10.2.1. Research Aim.....	83
10.2.2. Research Objectives.....	83
10.3. Utilization of Knowledge from the Course.....	84
10.4. Use of Existing Skills.....	85
10.5. Use of New Skills	86
10.6. Achievement of Learning Outcomes	86
10.7. Problems and Challenges Faced	87
10.7. Deviations	88
10.8. Limitations of the Research	88
10.9. Future Enhancements.....	89
10.9. Achievement of the Contribution to the Body of Knowledge	89
10.9.1. Problem Domain	89

10.9.2. Research Domain	90
10.10. Concluding Remarks.....	90
REFERENCES	A
APPENDIX A: IN-SCOPE, OUT-SCOPE AND PROTOTYPE DIAGRAM.....	H
APPENDIX B: CONCEPT MAP.....	I
APPENDIX C: GANTT CHART	J
APPENDIX D: SRS.....	K
APPENDIX E: DESIGN.....	P
APPENDIX F: IMPLEMENTATION.....	T
APPENDIX G: TESTING	Y
APPENDIX H: EVALUATION.....	AA

LIST OF TABLES

Table 1: Research Objectives.....	5
Table 2: Research Methodology	25
Table 3: Requirement Elicitation Methodologies	26
Table 4: Hardware Requirements	27
Table 5: Software Requirements.....	27
Table 6 :Risks and Mitigation.....	29
Table 7: Deliverables	29
Table 8: Stakeholder Analysis	33
Table 9: Requirement Elicitation Methodologies and Justifications	34
Table 10: Literature Review Findings	35
Table 11: Survey Findings	35
Table 12: Formal Interview Findings	39
Table 13: Summary of Requirement Findings.....	41
Table 14: UC2 Use Case Description	43
Table 15: MoSCoW Principles	44
Table 16: Functional Requirements	44
Table 17: Non- Functional Requirements.....	45
Table 18: SLEP Issues	47
Table 19: Design Goals.....	48
Table 20: Development Frameworks.....	55
Table 21: Libraries used for Implementation.....	56
Table 22 : IDE's used for Implementation	56
Table 23: Tools and technology Stacks	57
Table 24: Precision values of Test Data Sample	67
Table 25: Recall values of Test Data Sample	67
Table 26: F1 values of Test Data Sample	67
Table 27: Test Data Benchmarking Results.....	68
Table 28: Functional Testing Results	69
Table 29: Module and Integration Testing Results.....	70
Table 30: Evaluation Criteria.....	74

Table 31: Self-Evaluation Results	75
Table 32: Evaluators Count	76
Table 33: Evaluation Theme Defining.....	77
Table 34: Technical Expert Evaluation Opinions	77
Table 35: Domain Expert Evaluation Opinions.....	78
Table 36: Focus Group Evaluation Opinions.....	79
Table 37: Functional Requirements Evaluation.....	80
Table 38: Non-Functional Requirements Evaluation	81
Table 39: Research Objective Completion Status	83
Table 40: Course Modules Utilization.....	84
Table 41: Learning Outcome Achievements	86
Table 42: Problems and Challenges Faced	87
Table 43: SRS Formal Interview Thematic Analysis	L
Table 44: UC1 Use Case Description	N
Table 45: UC3 Use Case Description	N
Table 46: Evaluators Responses	AA

LIST OF FIGURES

Figure 1: Proposed Architecture Diagram (Self-composed)	20
Figure 2: ROUGE-N Equation (Lin, 2004).	22
Figure 3: ROUGE-L Equation (Lin, 2004).....	23
Figure 4: Rich Picture Diagram (Self-composed)	31
Figure 5: Stakeholder Onion Model	32
Figure 6: Context Diagram (Self-composed).....	42
Figure 7: Use-case Diagram (Self-composed).....	42
Figure 8: High-Level Architecture Diagram (Self-composed).....	49
Figure 9 : Level 01 DFD (Self-composed)	51
Figure 10: Level 02 DFD (Self-composed)	52
Figure 11: System Process Flowchart (Self-composed)	53
Figure 12: Technology Stack	54
Figure 13: Stopword Removal (Extractive).....	57
Figure 14: Word Frequency Table Generation (Extractive)	58
Figure 15: Sentence Score Generation (Extractive).....	58
Figure 16 : Threshold Identification Using Average Sentence Scores (Extractive)	58
Figure 17: Extractive Summarizer Function (Extractive).....	59
Figure 18: CNN Dataset before Translation	60
Figure 19:: CNN Dataset Translation	60
Figure 20: CNN Dataset after Translation	60
Figure 21: Hyper Parameter Tuning (Abstractive)	61
Figure 22: Training Arguments (Abstractive)	62
Figure 23: Combined Approach Function (Combined)	62
Figure 24 : SinDOC Home (UI Implementation)	63
Figure 25: SinDOC Summarizer (UI Implementation)	63
Figure 26: Training Results without Hyper Parameter Optimization	66
Figure 27 Training Results with Hyper Parameter Optimization	66
Figure 28: Web App Performance Report (Google Lighthouse).....	71
Figure 29: CPU and Memory Usage for Web Application.....	72
Figure 30: Web App Usability Report (Google Lighthouse).....	72

Figure 31: Web App Code Quality Report (CodeFactor)	73
Figure 32: SinDOC Feature Prototype Diagram (Self-composed)	H
Figure 33: Concept map (Self-composed)	I
Figure 34: Gantt Chart (Self-composed).....	J
Figure 35: SRS Survey -Page 1	K
Figure 36: SRS Survey -Page 2	L
Figure 37: SRS Survey -Page 3	L
Figure 38: SinDOC Home (Wireframe)	P
Figure 39; SinDOC Summarizer (Wireframe).....	Q
Figure 40: SinDOC Home (Prototype)	R
Figure 41 SinDOC Summarizer Before (Prototype).....	S
Figure 42: SinDOC Summarizer After (Prototype).....	S
Figure 43: Deployed Sinhala Translated Dataset.....	T
Figure 44: Remove Rows with Null Values	U
Figure 45: Removal of the word CNN.....	U
Figure 46: Removal of Unwanted Columns	U
Figure 47: Removal of URL links	U
Figure 48: Removal of HTML Tags	V
Figure 49: Removal of Consecutive Duplicate words	V
Figure 50: sinMT5 deployed model.....	W
Figure 51: Automatically Generated Training Arguments for each Trial	W
Figure 52: sinMT5-tuned deployed model	W
Figure 53: Abstractive summarizer function (Abstractive)	X
Figure 54: ROUGE scores on CNN test data (Extractive)	Y
Figure 55: ROUGE scores on BBC test data (Extractive).....	Y
Figure 56: ROUGE scores on CNN test data (Abstractive)	Y
Figure 57: ROUGE scores on BBC test data (Abstractive).....	Y
Figure 58: ROUGE scores on CNN test data (Combined)	Y
Figure 59: ROUGE scores on BBC test data (Combined).....	Y
Figure 60: GPU Usage while Training Model (Abstractive).....	Z

ACRONYMS

NLP	Natural Language Processing
SOTA	State Of The Art
T5	Text-To-Text Transfer Transformer
BERT	Bidirectional Encoder Representations from Transformers
BART	Bidirectional Auto-Regressive Transformers
PEGASUS	Pre-training with Extracted Gap sentences for Abstractive Summarization
RNN	Recurrent Neural Networks
Seq2Seq	Sequence to Sequence
TF-IDF	Term Frequency - Inverse Document Frequency
LSA	Latent Semantic Analysis
ROUGE	Recall Oriented Understudy for Gisting Evaluation
BLEU	Bilingual Evaluation Understudy
SSADM	Structured System analysis and Design Method
OOAD	Object Oriented Analysis and Design
API	Application Programming Interface
GPU	Graphic Processing Unit
CPU	Central Processing Unit
GUI	Graphical User Interface
UI	User Interface
UX	User Experience
SEO	Search Engine Optimization

CHAPTER 01: INTRODUCTION

1.1. Chapter Overview

This research will mainly focus on the Sri Lankan community who will have the difficulty in summarizing long Sinhala documents and getting the idea of it in simple terms. Sinhala language is the native language of all Sri Lankans. Many native language speakers and readers tend to miss the most important information in a document as they get bored or stop continuing to read a long document. This research focuses on providing summaries for such long Sinhala documents and help in conveying the most important context of a document to the people. Details of existing work, author's working schedules, project management methodologies to be used for this research are stated in the upcoming sections.

1.2. Problem Domain

1.2.1. Natural Language Processing (NLP)

Natural language processing is the process in which computers are trained for text and spoken related tasks that provide results as humans. (Geitgey, 2020). During recent times NLP has been a highly talked about and contributed research area. Some of the applications that are widely useful are machine translation, document translation, question answering and text generation (Geitgey, 2020). Industries such as healthcare, finance and customer service are highly benefited from NLP's use cases.

1.2.2. Low resource language

English related projects have been growing widely in the field of Natural Language Processing due to the extensive amount of data available in any type of platform. But languages like Sinhala are languages that do not have a wide resource available on the internet. These types of documents are known as low resource languages (Devlin et al., 2019). Even though there isn't much data on the internet, such a language is considered to be the primary language of a country. One such use case would be that there are around 21 million people within Sri Lanka who are considered to have Sinhala as their primary language and around 11.5 million internet users (Digital 2022: Sri Lanka, 2022), which means that even though it is a low resource language around the world it still has a lot of value within Sri Lanka. Summarization is one of the tasks that haven't been much focused when it comes to low resource languages (Hasan et al., 2021).

The author will be solving this problem by implementing a summarization solution for the low resource language Sinhala.

1.2.3. Document summarization

Document summarization is the task of collecting data from a long document content and summarizing it into shorter simpler sentences while maintaining the main content of the document (Gonçalves, 2021). Document summarization is mainly of two categories namely extractive and abstractive summarization. Extractive summarization is a methodology of extracting the most important sentences or words from a document and generates a summary while abstractive summarization uses the technique of paraphrasing the documents content while generating a summary (Carenini and Cheung, 2008). Hybrid approaches include both extractive and abstractive techniques for summarization (Hsu et al., 2018). During recent times researchers have been actively contributing on single document and multiple document summarization using both extractive and abstractive techniques (DeYoung et al., 2021).

1.3. Problem Definition

With the improvement of technology, the amount of text gathered being stored is large. Due to this a lot of people avoid consuming all data (Deshpande and Jahirabadkar, 2021). Different researchers have introduced different summarization techniques with the aim of solving the above problem. This helps people consume content with ease and less effort. Extractive, abstractive and combined summarizations have been applied for solving these problems. Extractive summarization techniques involve summary generation by extracting the most important sentences and concatenating them (Wolf et al., 2020). Abstractive summarization techniques generate summary by applying paraphrasing to the input text (Hsu et al., 2018). Combined summarization techniques use both the extractive and abstractive approaches to generate a structured summary. (Hsu et al., 2018)

Most of the summarization techniques have been tested and implemented for the English language. The main reason for this is due to the lack of datasets for other languages (Hasan et al., 2021). Sinhala language is one of the low resource languages that hardly has any datasets for any type of data science field. Machine translation (Guzmán et al., 2019), Text classification (Dhananjaya et al., 2022) are sections where NLP has been actively used for Sinhala language.

The author will be creating a novel approach for summarizing Sinhala documents which would contribute for both NLP and Sinhala domains. The limitation of datasets will be addressed and a combined approach for summarization will be implemented to provide better results.

1.3.1. Problem Statement

The author will be addressing the summarization task for Sinhala language domain which includes the creation of a new dataset that will help future researchers.

1.4. Research Motivation

Due to the recent growth in the field of NLP the author of the research felt interested in exploring its capabilities. With the recent developed projects on pre-trained large language models that have the capability of having multiple NLP tasks incorporated into them the interest grew even further. Researchers and developers fine-tuning such models and using them for their projects and solving research gaps (Lee, 2023) was identified by the author. The author also has a personal interest in the field of NLP. All the above-mentioned reasons motivated the author in conducting this research.

1.5. Research Gap

The author did come across many research gaps that were stated in previous literature work related to both summarization and Sinhala tasks. Some of the gaps that were mentioned in previous literature were identifying the best model for multilingual tasks (Sakhadeo and Srivastava, 2018), improving models for lingual summarization (Hasan et al., 2021), incorporating a question answer system within the summarization models (Zhang, Wei and Zhou, 2019). After careful consideration the author will be addressing the following gaps within the projects timeline.

- Create a Sinhala document summarizer tool with a combined approach for summarization. (Hasan et al., 2021)
- Create a new dataset for the summarizer model for Sinhala language. (Dhananjaya et al., 2022)
- Create a summarizer with no length limit for Sinhala input text. (Subramanian et al., 2020)

1.6. Contribution To The Body Of Knowledge

According to the research gap stated above, the author will be contributing both technically and domain wise. The technical and domain contributions are elaborated in the following topics.

1.6.1. Contribution to the Problem Domain

The research proposes to provide a summarized version of a long Sinhala document. This will help a lot of native Sri Lankans in understanding the context of a document in simple terms and will encourage reading more Sinhala documents. This will also be much less time-consuming since the user or consumer of the product will get a summarized version of the document. The impact of this can largely affect within the Sri Lankan community which will include a great contribution to the problem domain.

1.6.2. Contribution to the Research Domain

The app will be contributing into the low resource language domain as well as the summarization domain. The author will likely be building an application mainly focused on Sinhala as it is considered as a low resource language as it has very low data in the internet when compared to some of the very high resource languages like English (Hasan et al., 2021). The author will also be implementing an application that will be using best approaches of extractive and abstractive summarization techniques to build a combined model (Hsu et al., 2018). The author will also be introducing a new dataset that will be trained to achieve the goal. This will help with the issue of lack of dataset (Dhananjaya et al., 2022), which will help future researchers' contributions. With the above stated contributions, the author will be able to make a great impact to the research domain.

1.7. Research Challenge

The author came up with some challenges that the author would face while working on this research project. Some such research challenges are stated herewith.

- Identifying the right tools and techniques to build a model that provides a summary of the document in Sinhala.
- Researching and reviewing deep learning tools and techniques that produce results with better accuracy for low resource languages.
- Identifying the right dataset to achieve the aim of the research.

1.8. Research Questions

RQ1: What are the existing models that have been developed in document summarization for low resource languages?

RQ2: How have past researchers collected dataset for low resource languages in the field of Natural Language Processing?

RQ3: Will the use of pre-trained models be more effective than traditional methods in the case of document summarization for low resource language models?

RQ4: When it comes to document summarization, which is better, extractive or abstractive?

1.9. Research Aim

The aim of this research is to design, implement and evaluate a system that does document summarization for a Sinhala document or article and provide a summary of it.

To elaborate on the aim, the author will be creating a combined model for document summarization tasks that will be using a self-created dataset. The author will then test the model with real world scenarios and identify the best case and fulfill the research gap.

The author will be making the built system publicly available and will have the ability of running in machines with less power resources. This will encourage future researchers to use them in their work and make contributions. Necessary research and evaluation will be conducted by the author while working on the project.

1.10. Research Objectives

The author was able to come up with research objectives after reviewing the research aims and questions. Those research objectives are stated in the table below.

Table 1: Research Objectives

Research Objective	Description	Learning Outcomes	Research Questions
Problem Identification	This includes identifying a research problem that has enough potential to be solved. <ul style="list-style-type: none">• RO1: Research on document summarization and their usages	LO1, LO3, LO4	RQ1

	<ul style="list-style-type: none"> ● RO2: Research on summarization techniques used for low resource language domains. ● RO3: Identifying the importance of document or article summarization for native people of Sri Lanka 		
Literature review	<p>This includes in-depth research to assess existing SOTA works.</p> <ul style="list-style-type: none"> ● RO1: Review existing document summarization techniques ● RO2: Validate identified research gap. ● RO3: Identifying the importance of abstractive and extractive summarization. ● RO4: Identifying the usages of pre-trained models in the field of document summarization. ● RO5: Identifying datasets that would potentially help for the authors use case. 	LO1, LO4, LO8	RQ1, RQ2, RQ3
Data gathering and analysis	<p>This includes carrying out data gathering and analyzing them.</p> <ul style="list-style-type: none"> ● RO1: Gathering feedback on creating a model that summarizes documents or articles in Sinhala. ● RO2: Interview domain related people to understand the importance of carrying out this type of research. ● RO3: Gathering and evaluating response to understand requirements and create a better system 	LO3, LO4, LO6, LO8	RQ1,RQ3

Research design and implementation	<p>This includes the method of designing the architecture of the solution and implementing it.</p> <ul style="list-style-type: none"> ● RO1: Creating necessary diagrams to depict the functionality and the architecture of the system. ● RO2: identifying and preparing the necessary dataset for implementing the summarization model. ● RO3: Implementing the research model which includes the document summarization functionality. ● RO4: Creating a fully functional prototype of the system to use for real world scenarios 	LO2, LO5, LO7, LO8	RQ2, RQ3
Testing and evaluation	<p>This includes comparisons with existing models and testing application.</p> <ul style="list-style-type: none"> ● RO1: Providing the prototype to domain related experts to evaluate the experience and conduct black box tests. ● RO2: Doing testing to ensure relevant scores and metrics provide better results. ● RO3: Providing a detailed assessment of results to the community for future works. 	LO8, LO9	RQ2, RQ3

1.11. Chapter Summary

The chapter mainly focuses on the problem background. Initially an overview of the problem and research areas are stated. Then the research gaps are mentioned. The author then briefs about aims and objectives of the research. The in scope out scope and prototype diagram are also additional added in **APPENDIX A**.

CHAPTER 02: LITERATURE REVIEW

2.1. Chapter Overview

This chapter discusses the previous work that has been done related to the author's technological and problem domain. Critical review of literature, proposed architecture diagram, technological review and evaluation methods are elaborated in this chapter.

2.2. Concept Map

The concept diagram shows an overview of what the author's research looks like. It is attached in the **APPENDIX B**.

2.3. Research Domain

2.3.1. Summarization Techniques

Summarization techniques is where the content of a document or a block of paragraph is given to a model and the model tries to summarize the input by maintaining the most important contents of the document or paragraph. Automatic summarization is one of the most important NLP tasks that has been growing recently. The main reason for it is the availability of storing a lot of data digitally. Users tend to avoid the main context of whatever they read as they get bored of consuming a lot of information at the same time. Researchers make contributions to the summarization field mainly because of this reason. Low resource languages are one of the upcoming research areas that researchers want to get their hands on in an experiment on various methods of producing the best model (Guzmán et al., 2019).

Summarization can be broken down into two main domains.

- Text summarization
- Document summarization

2.3.1.1. Text Summarization

The process of generating a single line summary from a block of paragraph or a set of sentences is called text summarization. Highly researched areas for such summarization techniques will be like moral generation, headlines generation and statement generations.

Text summarization is highly challenging as it needs to identify the most important contents of a large input text and generate only one sentence (Zhou and Hovy, 2004). Various methods have been applied for text summarization during recent years. Some such are extractive approaches (Filippova et al., 2015), abstractive approaches (Chopra, Auli and Rush, 2016), keywords

clustering approaches (Zhou and Hovy, 2004) and other deep learning techniques. The lack of resources for low resource languages than high resource languages make it a difficult area of exploration (Joshi et al., 2020).

2.3.1.2. Document Summarization

On the other hand, document summarization includes the generation of a summarized text block for an input document. This does not limit to a single line of text. The work on document summarization has been growing recently, since the amount of data on the internet is very high and almost all the documents are rich in text (Hasan et al., 2021). Document summarization also includes several types of summarization techniques namely extractive (Subramanian et al., 2020), abstractive (Subramanian et al., 2020), LSA techniques (Carenini and Cheung, 2008), page rank algorithms (Carenini and Cheung, 2008) and other deep learning techniques (Chopra, Auli and Rush, 2016). Document summarization includes two types of summarizations. The two types of document summarizations are stated as follows.

- **Single document summarization -**

This is where the users will have to input some text only from a single document. The summary will be generated from that single document (Goldstein et al., 2000). News summarization and article summarization is a highly considered use case of single document summarization (Deshpande and Jahirabadkar, 2021).

- **Multi- document summarization -**

This is where the users will have to input text from multiple documents. The model identifies the most important content from each document and connects them to generate a combined summary (Goldstein et al., 2000). This type of document summarization is still being highly researched as the SOTA methods also do not provide that good of results (DeYoung et al., 2021). Research document summarization is a highly considered domain use case (DeYoung et al., 2021).

The author will also be choosing this domain of summarization as it is a more recent active area of research.

2.3.2. Sinhala Language

Sinhala language is considered to be one the very low resource languages when it comes to any machine learning or deep learning related projects. A very small number of contributions have been made for Sinhala and NLP related domains. Some of the known contributions are text

classification (Dhananjaya et al., 2022), sentiment analysis, optical character recognition (Rathnasena et al., 2018), hate speech detection (Sandaruwan, Lorensuhewa and Kalyani, 2019), frequency sentence summarization methods (Rathnasena et al., 2018). Even though less projects have been done related to Sinhala, the language starts to grow widely within the internet. A lot of Sri Lankan people consume data in Sinhala as it is their primary language. A count of 11.3 million people out of 21 million around Sri Lanka alone uses the internet (Digital 2022: Sri Lanka, 2022). This means that there are a great number of people who will benefit from applications that are optimized for Sinhala language.

2.3.3. Why Document Summarization for Sinhala Language

As mentioned above there is almost 55% of the Sri Lanka population consuming the internet with an average of 1.5 million users daily (Digital 2022: Sri Lanka, 2022). Most of the users consume data in Sinhala language as it is their primary language. This shows that there is a possible number of consumers for applications related to Sinhala language. The survey result gathered in the software requirement specification **chapter** also shows that there are a lot of Sri Lankans who read Sinhala related documents. As the digital area starts to grow, the information stored starts to increase and the level of consumption reduces (Hasan et al., 2021). This has already begun with high resource languages such as English and French (Deshpande and Jahirabadkar, 2021). This is also predicted to be happening to low resource languages soon. As Sinhala is a low resource language (Guzmán et al., 2019) it could be said that due to the increase of data around the internet Sri Lankan people will also stop consuming long Sinhala documents and will be expecting to have shorter simplified versions of them (Hasan et al., 2021). Models are being implemented for summarizing documents for low resource languages but limitations such as character lengths, scarcity of data, grammatical inconsistencies reduce their level of accuracy but by time they will be improving (Dhananjaya et al., 2022). This research will be mainly focused on Sinhala document summarization, and it will be the first attempt of doing so. The author identifies the importance of creating such an application as the author foresees that this could be one of the future research areas that would be addressed within the Sinhala research space and provides a great contribution within the Sri Lankan and Sinhala community.

2.4. Existing Work

2.4.1. Traditional Approaches

2.4.1.1. NLP for Sinhala Language

This section will discuss the most relevant projects that have been implemented for the Sinhala language domain using Natural Language Processing techniques.

Dhananjaya et al. (2022) proposes a new model called SinBERT for text classification tasks. The model has been created by transfer learning RoBERTa (Liu et al., 2019). Additionally, the authors of this research have introduced two datasets that are specific for Sinhala text classification tasks (Dhananjaya et al., 2022). Compared to previous state-of-the-art methods their model performs best for classification tasks due to their larger number datasets with unique values (Dhananjaya et al., 2022). The model has been trained with four main classification data namely sentiment analysis, writing style classification, news source classification and news categorization whereas writing style and news source classification datasets were introduced by the authors (Dhananjaya et al., 2022). SinBERT-small and SinBERT-large were introduced in this paper however the authors recommend using SinBERT-small due to the less accuracy of the large model. They have stated that the issue of less accuracy is because of the less amount of data available for Sinhala that needs to be passed into a higher number of hidden layers within the model (Dhananjaya et al., 2022) . The authors are interested in testing the models for other NLP tasks and improve the model by feeding more data into it.

Rathnasena et al. (2018), proposes a method of using computer vision and summarization methods to store and search through old Sinhala books. The method includes Optical Character Recognition which extracts text information through images and then uses extractive summarization techniques to extract the most important information of the document. Due to the improvement of technology of storing old preserves the author identifies that there are no such methods for storing old sinhala books. Due to that the author decides to build a model that automates the process of recognizing characters from old Sinhala books and generating summaries of them to store for future readings (Rathnasena et al., 2018). The lack of sinhala corpus and the smaller number of historical words stored tend to reduce the accuracy of the model by some slight percentage. Additionally, the author also recommends abstractive summarization techniques for future work because of the paraphrasing and restructuring it could

provide for a document. (Rathnasena et al., 2018). The use of deep learning techniques and RNN's are recommended for future works (Rathnasena et al., 2018).

As an overall there have been a very few numbers of studies related to Sinhala language and summarization. As shown above there has been only one such existing work and that also includes only an extractive summarization technique (elaborated in next topic).

2.4.1.2. Extractive Summarization

This section will discuss the most relevant projects that have been implemented for the extractive summarization.

Deshpande and Jahirabadkar (2021) research focus on extractive summarization methodologies done using lexical chaining approach and pre-trained model BERT (Devlin et al., 2019). The model proposed is trained using Devanagi script data which includes Hindi Language (Deshpande and Jahirabadkar, 2021). The author has tested the lexical approach initially and then trained the dataset for the BERT models. The author provides some details on the outputs but does not include any test results, so it is hard to confirm if it performs better than the SOTA approaches. The author highlights that low resource languages are an important domain that should be considered for text summarization (Deshpande and Jahirabadkar, 2021).

Zhang, Wei and Zhou (2019) proposes a method of using a pre-trained model for better extractive summarization techniques. The model is fine-tuned on top of BERT (Devlin et al., 2019). The inclusion of hierarchical transforms to BERT implements their model HIBERT (Zhang, Wei and Zhou, 2019). The model successfully surpasses the SOTA models as demonstrated using the necessary evaluation methods. The authors state that the model could be improved with a larger dataset (Zhang, Wei and Zhou, 2019). They would also like to take the model to the extent of having the possibility of including a question answer system within the documents (Zhang, Wei and Zhou, 2019). Additionally, they would also like to improve the architecture to provide better results.

Sakhadeo and Srivastava, (2018) propose a method of extractive summarization that includes word frequency base approaches and entity graph generation. The authors have conducted both qualitative and quantitative approaches of testing to evaluate the performance of their model. They were able to identify that the quantitative results show that their model outperformed the SOTA methods while their human evaluation or qualitative methods show that most of the human did highly prefer their models' generated summaries when compared to previous SOTA

models (Sakhadeo and Srivastava, 2018). The authors were able to solve limitations of both approaches word frequency and entity relationship generation by creating a hybrid model that includes both the approaches (Sakhadeo and Srivastava, 2018). Identifying a suitable normalized function and improving the model for multilingual tasks are future improvements that the authors have stated.

As stated above these are some of the active researches of extractive summarization. Since an extractive method will be used for this research, the author will be using the most suitable model for the implementation. The author will be discussing the choice of method and its justifications in the upcoming chapters.

2.4.1.3. Abstractive Summarization

This section will discuss the most relevant projects that have been implemented for the abstractive summarization.

Hasan et al. (2021) is a work on abstract summarization tasks that introduces a multilingual dataset of 44 languages. The research was mainly focused on low-resource languages and its aim was to build a model that can be used for multiple languages in the field of summarization. It is also stated to have the largest abstractive summarization dataset for multiple low resource languages (Hasan et al., 2021). The model fine-tunes a pre-trained model mT5 (Mishra, 2021) in order to achieve its aim. The model built is openly available for contributions. Even Though multilingual models have been used in NLP tasks previously this is the first model to implement it in the context of abstractive summarization (Hasan et al., 2021). However due to the computational power the model had to train the model only for five languages from the dataset. They have also stated that due to the less amount of data fed into the model for certain languages the model provides a less evaluation score and the issue of stemming in certain languages is also considered to be a limitation (Hasan et al., 2021). The research also opens future work on lingual summarization (Zhu et al., 2019).

Timalsina, Paudel and Shahi (2022) proposes an abstractive summarization technique for Nepali text summarization tasks. The authors had to create a dataset from scratch. They have stated that inorder to create the dataset they have scraped data from a Nepali news online website (Timalsina, Paudel and Shahi, 2022). They propose a method of building a model on a RNN with attention techniques. They have clearly stated that the Nepali language method of implementation is comparatively different to languages like English (Timalsina, Paudel and

Shahi, 2022). The authors claim to be the first to implement an abstractive summarization model for Nepali language (Timalsina, Paudel and Shahi, 2022). The dataset created is not publicly available but could be gathered upon request. The scarcity of data for a low resource language such as Nepali has been a main limitation of their research. The usage of bi-directional encoders and pointer generation mechanisms (Timalsina, Paudel and Shahi, 2022) are highly recommended for future research by the authors of this research.

Wan and Bansal (2022) propose the idea of presenting summaries that are accurate and factually correct. The authors were able to use PEGASUS (Zhang et al., 2020) for fine-tuning their datasets and provide improved results during evaluation. They have stated that the sentence errors and token errors generated within the summary have been reduced when compared to SOTA methods (Wan and Bansal, 2022). They have also stated that improvement in meaning doesn't always increase the ROGUE scores (Wan and Bansal, 2022). They would like to improve the model more so that the ROGUE values also will improve, and it can be used in real world applications (Wan and Bansal, 2022).

The above stated researches are some of the active areas of research in abstractive summarization. Since the author will be implementing an abstractive model too, the best available model will be used.

2.4.1.4. Hybrid Summarization

This section will discuss the most relevant projects that have been implemented for the hybrid summarization which involves the combination of both extractive and abstractive approaches.

Hsu et al. (2018) proposes a research approach of creating a hybrid summarization model with inconsistency loss that includes extractive and abstractive techniques. The model creates a sentence level extractive method and a word level abstractive method for text summarisation (Hsu et al., 2018). For the model building the authors have used the combination of the best previous models of extractive and abstractive summarization techniques (Hsu et al., 2018). The authors claim that they have outperformed the SOTA methods with their unified model. To prove the performance of the model the authors have done both automatic and human evaluation. Eventhough though the author has not specifically mentioned future work (Hsu et al., 2018). This type of an approach havent been used for low - resource languages which makes it a potential area of research.

Veenadhari and Bharathi (2022) has proposed a method of hybrid summarization that uses Seq2Seq architecture with RNN. The authors initially planned on implementing it in the legal research domain but due to the scarcity of dataset they had to move onto news summary dataset (Veenadhari and Bharathi, 2022). They had stated that the technological concept can be used for any domain if the necessary data is fed into the model. The collection of a domain related dataset and the use of pointer generator approaches are stated as future works of the research (Veenadhari and Bharathi, 2022).

Tretyak and Stepanov (2020) research proposes a combined approach of extractive and abstractive techniques for summarizing scientific documents. The authors had used BART (Lewis et al., 2019) for the extractive summarization process and BERT (Devlin et al., 2019) for the abstractive summarization process. The evaluation metrics show that the proposed model outperforms the SOTA methods including the only abstractive model and only extractive model too (Tretyak and Stepanov, 2020). As a future work the author would like to try the model on various other domain datasets (Tretyak and Stepanov, 2020).

These are some of the recent researches on hybrid summarization. Since the author will be implementing a hybrid model the author will choose the most suitable extractive and abstractive models to build the hybrid model.

2.4.2. Other approaches

Apart from the traditional approaches stated above the following research follows some different approaches to solve the problem of summarization tasks as well as low resource language tasks.

Fabbri et al. (2019) introduce a large-scale multi-document news summarization dataset in their research. They are the first to introduce a multi-document dataset to the community. They additionally build a model using extractive summarization techniques to evaluate their dataset on various metrics. The research uses pointer generator networks for extractive summarization. The authors conducted both automatic and human evaluation. They have stated that human evaluation is more important than automatic evaluation when it comes to summarization (Fabbri et al., 2019). Exploring longer documents beyond concatenations (Fabbri et al., 2019) is a future work stated by the authors.

Liu and Lapata (2019) is research that focuses on fine-tuning a pre-trained model and using it for both extractive and abstractive approaches separately. The model uses BERT (Devlin et al., 2019) for the task of transfer learning. The BERT model considers input information as tokens

whereas this model considers information as sentences which is important for summarization tasks. The model involves a two state fine-tuning which is initially fine-tuned for the extractive stage and then passed on to the abstractive stage (Liu and Lapata, 2019). The model also uses three datasets for fine-tuning. Since the research is mainly focused on summarization, the authors would like to use it for language generation for future works.

In Lee et al. (2022), the authors work on researching the use of pre-trained models for low resource languages and machine translation tasks. The authors identify the usage of mBART (Liu et al., 2020), a pre-trained model that performs well on their use cases. The authors have gathered some of the high-level resource languages and low resource languages for evaluating their system. The scarcity of data on low resource languages is pointed out as a limitation (Lee et al., 2022). The researchers say that if more of the data related to the specific language can train the model better and provide more accurate results.

2.4.3. Summary of existing work

The traditional approaches and other approaches stated in previous topics are summarized in the table below.

Citation	Technology	Improvement	Limitations/ Future work
(Dhananjaya et al., 2022)	RoBERTa	<ul style="list-style-type: none"> Introduces SinBERT-small and SinBERT-large. Introduced two new datasets for model training namely writing style and news source classification. The first model fine-tuned for Sinhala language to do text classification tasks accurately. The model is publicly available for all future researchers. 	<ul style="list-style-type: none"> Interested in exploring other NLP tasks with models. Lack of dataset makes SinBERT-large less accurate, so authors are interested in improving the amount of data to train the model.

(Rathnasena et al., 2018)	OCR, Sentence level extraction	<ul style="list-style-type: none"> • The first and only model that attempts Sinhala summarization as of authors knowledge. • Provide good accuracy in Sinhala character recognition. 	<ul style="list-style-type: none"> • Authors recommend abstractive summarization. • Deep learning and RNN is also recommended for future summarization tasks. • The lack of dataset in Sinhala language is also addressed
(Deshpande and Jahirabadkar , 2021)	Lexical chaining, BERT	<ul style="list-style-type: none"> • Model has been implemented from a Devanagi script dataset which is having a low resource language. 	<ul style="list-style-type: none"> • Summarization techniques need to be given more attention for low resource languages. • Lack of a good dataset
(Zhang, Wei and Zhou, 2019)	BERT	<ul style="list-style-type: none"> • Pretrained model implementation for low resource languages. 	<ul style="list-style-type: none"> • Build a question answering system. • Improve model architecture
(Sakhadeo and Srivastava, 2018)	Word frequency, Entity relationship graphs	<ul style="list-style-type: none"> • Solved limitations of both word-frequency and entity relationship graphs by combining them. 	<ul style="list-style-type: none"> • Improve model for multilingual tasks. • Identifying a better normalized function for the model
(Hasan et al., 2021)	mT5	<ul style="list-style-type: none"> • Fine-tuned T5 model for 44 low resource languages. • First abstractive model to be trained on low resource 	<ul style="list-style-type: none"> • Models are only mainly focused on five languages due to computational power.

		<p>languages.</p> <ul style="list-style-type: none"> • Authors extracted and created dataset by themselves which is available openly. 	<ul style="list-style-type: none"> • Lack of dataset for certain languages limits training to a lesser ROUGE score. • Lingual summarization is not a part of their model.
(Timalsina, Paudel and Shahi, 2022)	RNN with attention	<ul style="list-style-type: none"> • First abstractive summarization model for Nepali text summarization. • Dataset was created by themselves by scraping data from an online news website and creating human-like summaries. 	<ul style="list-style-type: none"> • The scarcity of a rich dataset • Usage of bi-directional encoders and pointer generator networks are recommended as a future work.
(Wan and Bansal, 2022)	PEGASUS	<ul style="list-style-type: none"> • Fine-tuned model to generate factually correct summaries. • Reduces fake generated sentence and token errors 	<ul style="list-style-type: none"> • Improve model to get better ROGUE values. • Use model for real world applications
(Hsu et al., 2018)	Sentence - level, Word-level extraction with inconsistency loss.	<ul style="list-style-type: none"> • The best of both extractive and abstractive models were used to build this hybrid model. 	<ul style="list-style-type: none"> • Application of the model to low resource languages

(Veenadhari and Bharathi, 2022)	Seq2Seq with RNN	<ul style="list-style-type: none"> • First hybrid model to be implemented using Seq2Seq with RNN 	<ul style="list-style-type: none"> • Scarcity of dataset in legal domain • Use of pointer generator networks.
(Tretyak and Stepanov, 2020)	BART, BERT	<ul style="list-style-type: none"> • Outperforms BERT and BART model ROUGE individual scores when combined. 	<ul style="list-style-type: none"> • Try on other domain datasets
(Fabbri et al., 2019)	Pointer generator networks	<ul style="list-style-type: none"> • Introduces the largest multi-document summary dataset 	<ul style="list-style-type: none"> • Exploring longer documents beyond concatenation.
(Liu and Lapata, 2019)	BERT	<ul style="list-style-type: none"> • Two stage fine-tuning. First for the extractive model and then for the abstractive model. • Uses three datasets for fine-tuning 	<ul style="list-style-type: none"> • Will like to improve model for language generation tasks
(Lee et al., 2022)	mBART	<ul style="list-style-type: none"> • Fine tune model for a low resource language dataset. 	<ul style="list-style-type: none"> • Scarcity of data for those low resource languages is a limitation.

2.5. Technological Review

2.5.1. Proposed Architecture for the Sinhala Document Summarization Application.

The following diagram shows how the author will be implementing the application that summarizes a Sinhala model. The user will have the ability to get a summary of the Sinhala document on any model as preferred. To summarize the functionality of the application, the user

will have to initially input the document content and then choose the type of summarization they would like to get. The flows of summarization are listed as follows.

- If the user chooses **extractive summarization** (Sakhadeo and Srivastava, 2018) they will get a summary of the document from extracted sentences.
- If the user chooses **abstractive summarization** (Timalsina, Paudel and Shahi, 2022) they will get a summary of the document with human-like paraphrases.
- If the user chooses **combined summarization** (Hsu et al., 2018) they will get a summary of the document with human-like paraphrases that contain the most important sentences.

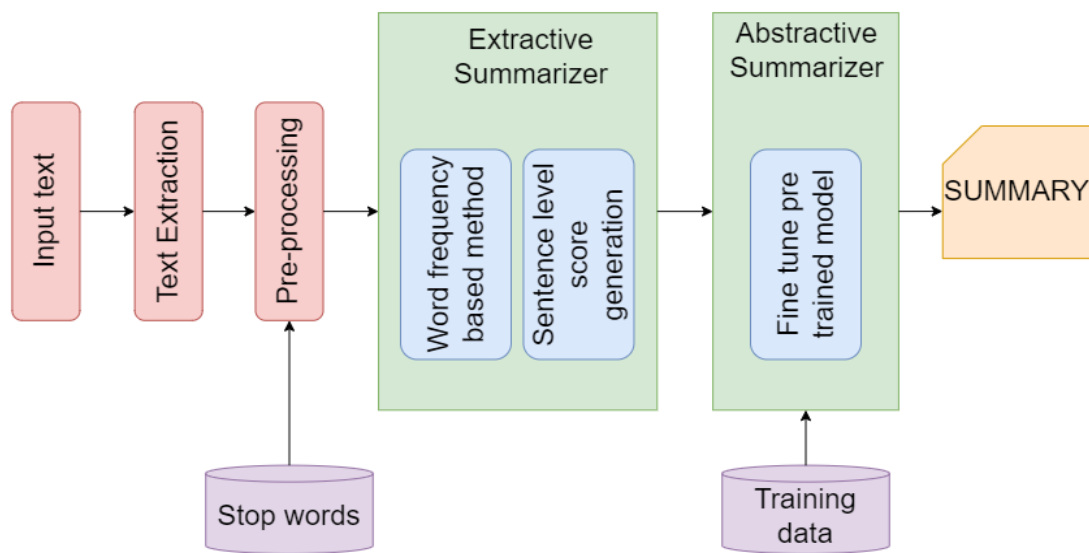


Figure 1: Proposed Architecture Diagram (Self-composed)

2.5.2. Algorithmic Approaches

This technique was initially and still used only for extractive summarization. The technique involves word frequency and sentence scoring algorithmic approach to extract the most important sentences from a document (Allahyari et al., 2017). TF-IDF and word probability are the most known frequency-based algorithms. LSA is another method that forms a word matrix and compares the cosine similarity between the words and to get the most used sentence and generate the summary (Steinberger and Ježek, 2004). Graph based algorithms are another algorithmic method. This is where the document's words are mapped as a graph to identify the most important words in it (Mihalcea, 2004). Popular graph methods include PageRank and TextRank algorithms (Mihalcea, 2004).

2.5.3. Machine Learning Approaches

When it comes to summarization models were initially built using machine learning. Most explored areas using machine learning are semantic based approaches and query-based summarization (Rahul, Adhikari, and Monika, 2020). Other than that, some of the researchers used Naive Bayes, Support Vector Machines, K-means clustering and such to build text summarization models.

Both extractive and abstractive summarization initially used machine learning techniques until deep learning came into play within the NLP domain and provided more accurate and better results.

2.5.4. Deep Learning Approaches

Deep learning approaches have been widely popular in the field of text summarization for a couple of years. Researchers use many different approaches to deep learning in order to achieve the best result for both extractive and abstractive models.

Researchers use a lot of pre-trained large language models such as T5 (Mishra, 2021), BART (Lewis et al., 2019), BERT (Devlin et al., 2019), PEGASUS (Zhang et al., 2020) and transfer learn according to their domain. The use of such models has been very efficient as they have pre-trained on large datasets specific for NLP tasks.

The Transformers method is the most popular approach that is available for document and text summarization (Divya et al., 2020).

2.5.5. Dataset Preparation

CNN Daily-mail, Xsum, DUC, Gigaword are some of the well collected datasets for summarization tasks. Most of the large language models have been trained on these datasets. But all of these datasets are primarily focussed on the English language. When it comes to low resource datasets, previous researchers mostly create their own datasets by scraping them from websites or manually creating them (Timalsina, Paudel and Shahi, 2022).

Google's translate API has been highly improving by time when it comes to translation tasks. Researchers even state that in many cases the translation includes good grammatical correction. Using real data is the most appropriate method compared to translation but previous researchers who have worked on low resource languages have stated various other ways of gathering data (Deshpande and Jahirabadkar, 2021). In Fact, they prove that if the model could score well for

such a dataset it will work very well on a well-made dataset (Timalsina, Paudel and Shahi, 2022).

2.5.6. Data Pre-processing

Data processing is one of the most important steps involved when preparing the dataset for training. When it comes to summarization tasks it is an important matter that needs to be considered.

Removal of duplicate words, punctuations, stopwords, URL's, HTML tags are common preprocessing methods in text summarization (Divya et al., 2020). When it comes to the Sinhala language it has its own limitations. Certain pre-trained models now include the ability of Sinhala tokenizer, but they don't provide the best results at all times (Dhananjaya et al., 2022). By time it's assumed to grow and evolve.

2.6. Evaluation Methods on Document Summarization

Summarization techniques include mainly two types of evaluation techniques. Namely,

- Automatic evaluation
- Human evaluation

2.6.1. Automatic Evaluation

Automatic evaluation is where the computer generates cores in order to evaluate the performance of a model. `when it comes to summarization evaluation techniques there are many techniques but the most used and tested on are the following two methods.

- ROUGE (Recall Oriented Understudy for Gisting Evaluation)
- BLEU (Bilingual Evaluation study)

ROUGE metric

This metric is most used for evaluating summarization and translation tasks. When it comes to summarization the metric evaluates on how well a generated summary is related to actual summary. The metric uses n-gram methods to evaluate the performance of the model. The n-grams define the number of word combinations. As an example, if the n-gram is 1 (ROUGE-1) that means it will identify the one-word combinations between the actual summary and generated summary (Lin, 2004). The equation for ROUGE-N is stated below.

$$\text{ROUGE-N}_{\text{single}}(\text{candidate}, \text{reference}) = \frac{\sum_{r_i \in \text{reference}} \sum_{\text{n-gram} \in r_i} \text{Count}(\text{n-gram}, \text{candidate})}{\sum_{r_i \in \text{reference}} \text{numNgrams}(r_i)},$$

Figure 2: ROUGE-N Equation (Lin, 2004).

Another important value is ROUGE-L. This identifies the longest matched word sequence. The equation for ROUGE-L is stated below.

$$\text{ROUGE-L}_{\text{single}}(\text{candidate}, \text{reference}) = \frac{(1 + \beta^2)R_{\text{lcs}}(\text{candidate}, \text{reference})P_{\text{lcs}}(\text{candidate}, \text{reference})}{R_{\text{lcs}}(\text{candidate}, \text{reference}) + \beta^2 P_{\text{lcs}}(\text{candidate}, \text{reference})},$$

Figure 3: ROUGE-L Equation (Lin, 2004).

BLEU metric

The metric measures the number of words from the generated summary that are matching with the actual summary compared to the total number of words from the summary (Papineni et al., 2002). It also compares the computer-generated summary with the actual summary. The equation for this metric is stated below.

Blue Score = Brevity Penalty X Geometric Average Precision Scores. (Divya et al., 2020)

2.6.1. Human Evaluation

Even though automatic evaluation methods are used to test the performance of a summarization model most of the researchers say that human evaluation (Sakhadeo and Srivastava, 2018). Both qualitative and quantitative methods can be used for human evaluation. As humans are the ones who could give the best review on a generated summary, human evaluation is highly recommended by previous researchers (Lee et al., 2022). It is also important to know that automatic evaluation includes a lot of limitations when it comes to low resource language summarization (Timalsina, Paudel and Shahi, 2022). This reason alone makes human evaluation a high priority.

2.6.3. Benchmarking

There haven't been models that have specifically been implemented for Sinhala document summarization. Only one researcher has attempted to provide a solution for low resource languages. Due to time and resource limitations, they also did train for the top ten of those languages which does not include Sinhala language (Hasan et al., 2021). One of the GitHub

contributors did use the dataset introduced by Hasan et al. (2021) and have stated their test results. The comparison of our model and their model results are shown in the **Chapter 8.5**. But it is important to note that their specific process of evaluation is not mentioned explicitly.

2.9. Chapter Summary

The chapter was mainly focused on critical evaluation of existing literature work on the author's research and problem domain. First the research domain and its impact are discussed. Then the existing work on summarization and Sinhala language are critically reviewed. The technology used in previous research was then reviewed and discussed. Finally, the evaluation metrics for summarization models were explained.

CHAPTER 03: METHODOLOGY

3.1. Chapter Overview

This chapter includes the methods that will be used by the author to conduct the research. The approaches, time schedule, resources required, risks that could happen are all elaborated within this chapter.

3.2. Research Methodology

Table 2: Research Methodology

Research philosophy	Pragmatism will be chosen by the author for this research project. This applied research will include involvement in going through different models and methods.
Research approach	Deductive approach will be used in this research since the author will be involving existing work references to come up with the proposed solution
Research strategy	Surveys, interviews and brainstorming strategies will be used by the author while working on this project.
Research choice	Mixed method will be used in this project since it involves analyzing of both qualitative and quantitatively measured data
Time zone	Cross-sectional time zone method will be used to conduct this project since it will be involving a particular moment from a time frame
Research techniques/ Data collection methods	Interview, survey results will be used to collect data relevant to the project. Datasets that will be used for this project are publicly available for usage.

3.3. Development Methodology

3.3.1. Requirement Elicitation Methodology

The author will be collecting data and analyzing it to build a better-quality system as proposed. To collect them the author will be using requirement elicitation methods as stated in the table below.

Table 3: Requirement Elicitation Methodologies

Requirement elicitation methodology	Justification
Literature Review	This will help the author to collect and analyze existing works and their relevant technologies
Interview	Interview domain and technology experts will help the author to collect valuable information of their expertise in building a quality system.
Survey / Questionnaire	Collecting and analyzing the statistical data from the user will be helpful to gather data on improvements to the system

3.3.2. Design methodology

The SSADM design methodology will be used to design the system. The decision was made by the author due to the following reasons. The SSADM methodology will provide the best design approach in dividing the stages, modules, workload, and tasks in developing the project. By using this methodology, the author will be able to develop a system that provides a more effective and efficient way to develop a quality system in a better way.

3.4. Project Management Methodology

Prince 2 Agile project management methodology will be followed throughout the project. The reason for choosing this methodology among several choices is because the Prince 2 agile methodology will allow the author to focus on completing and delivering the project on time. This will also help the author to provide a better-quality system by the end of the time frame allocated for the project. To further elaborate, Prince 2 Agile methodology will help the author to

manage tasks, be self-organized, have fixed dates and deadlines and be agile. By following this project management methodology, the expected results of the project could be achieved during the timeframe without any hassle.

3.4.1. Resource Requirements

3.4.1.1. Hardware requirements

The table below states all hardware resources that will be used during this project and their justifications.

Table 4: Hardware Requirements

Requirement	Justification
Core i7 8th generation	This will provide the processing power required for the project
8GB RAM	This will be used to save and load all datasets that will be used in the project
GPU	This will be used to train all deep learning models related to this project
32GB / above of storage space	The pace will be used to save all files related to the project (code, test files, documents)

3.4.1.2. Software requirements

The table below states all software resources that will be used during this project and their justifications.

Table 5: Software Requirements

Requirement	Justification
Windows 10 OS or above / Linux	A 64-bit version of windows can be used for this project. This will manage all the heavy processes involved in the project.

Python/ R	The languages that will be used for all machine learning and deep learning tasks
Flask/Django	The backend of the prototype application could be developed using these frameworks
HTML & Bootstrap	The frontend of the prototype application could be developed using these frameworks
Google Collab	With GPU power in the cloud this can be used to test and work with the deep learning models
TensorFlow	This framework can be used for deep learning concept developments
Figma	The tool will be used for the designing of UI and UX elements of the prototype
Google Workspace	This workspace will be used to manage and store all the documents relevant to this project
Ms. Office	This will be used for documentation of reports and creation of slides to be presented
Zotero	This tool will be used to maintain all citations and references

3.4.1.3. Skill requirements

The authors skill requirements that will be needed for this research are stated as follows.

- Creation of dataset using translation API's
- Ability of transfer-learning and fine-tuning pre-trained models
- Build a web application that includes the summarization system.

3.4.1.4. Data requirements

The datasets that will be used for this research project are publicly available in Kaggle and Huggingface for free usage.

3.4.2. Risk management

The following table consists of risks that the author might face while working on this project and the steps that author would follow to overcome those risks.

Table 6 :Risks and Mitigation

Risk	Probability of Occurrence	Magnitude of the loss	Mitigation Plan
Changes in requirement	3	5	Following the Prince 2 Agile methodology for management and the prototyping methodology for design will make the author stay focused on the requirements
Limited hardware resources	5	5	Online platforms like Google Collab, Kaggle can be used since they use GPU powers from the cloud
Knowledge on domain	3	5	Doing a thorough research on existing domain work

2.4.3. Schedule

2.4.3.1. Deliverables

Table 7: Deliverables

Deliverable	Date
Project Proposal	09 th Nov 2022
Literature Review Document	27 th Oct 2022

Software Requirement Specification	24 th Nov 2022
System Design Document (Proof of Concept)	23 rd Dec 2022
Project Specifications Design and Prototype (PSDP)	16 th Feb 2023
Test and Evaluation Report	23 rd Mar 2023
Draft Project Reports	30 th Mar 2023
Final Project Report (Thesis)	02 nd May 2023

2.4.3.2. Gantt chart

The Gantt chart for this research project is attached in the **APPENDIX C**.

2.5. Chapter Summary

This chapter includes the methods the author would follow in order to successfully complete the research project proposed. The development methodology, the choice of project management methodology and resource requirements are all discussed.

CHAPTER 04: SOFTWARE REQUIREMENT SPECIFICATION

4.1. Chapter Overview

This chapter will be discussing the stakeholders of the system, use cases and requirements of the proposed system. It will also include the methods of research conducted by the author to gather necessary system requirements.

4.2. Rich Picture Diagram

The following rich picture diagram provides a helicopter view of the system and interactions that could happen between the system and all other parties. It will help the author in understanding the stakeholders and requirements that are necessary for this solution.

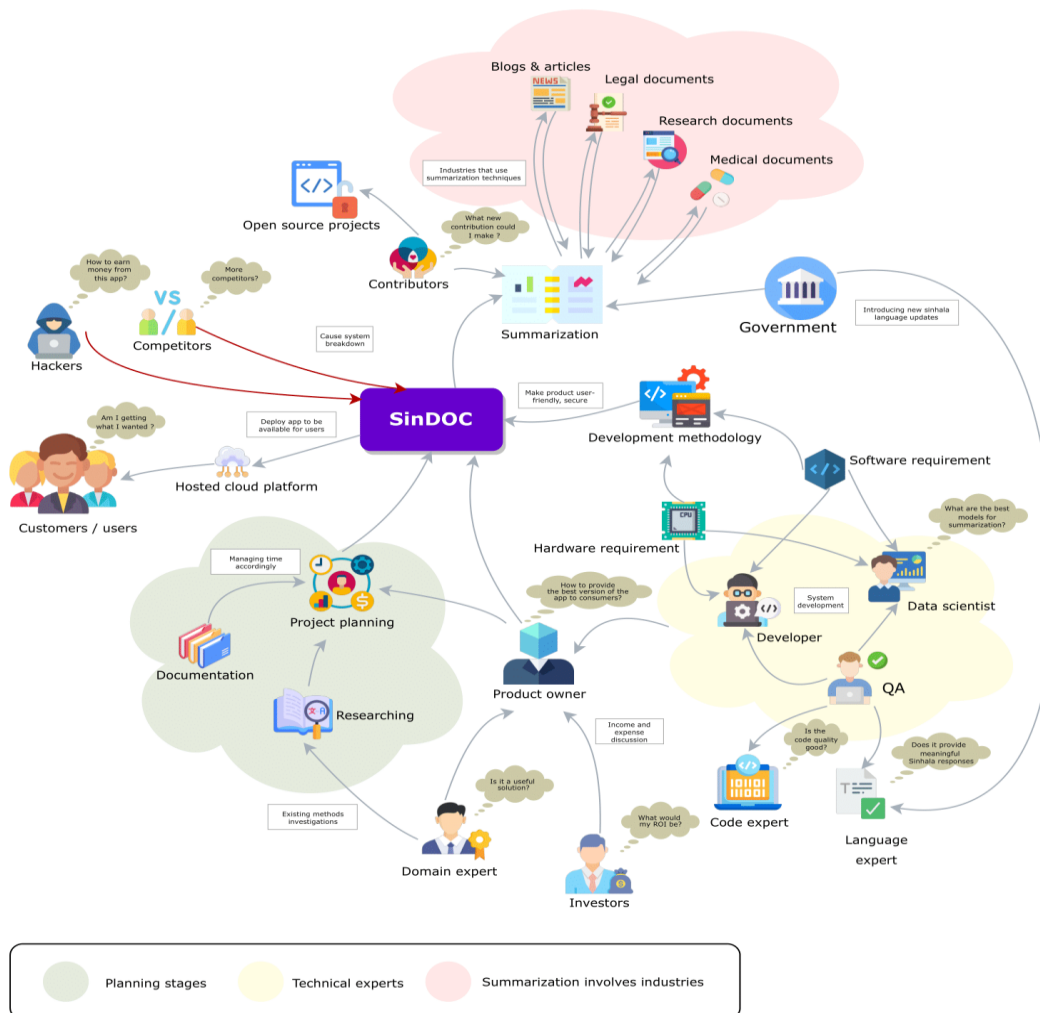


Figure 4: Rich Picture Diagram (Self-composed)

4.3. Stakeholder Analysis

4.3.1. Stakeholder Onion Model

The stakeholder onion model attached below helps the author identify all stakeholders that will affect the system during each stage and under different environments. This will also provide an understanding of all positive and negative stakeholders that will be included within the system.

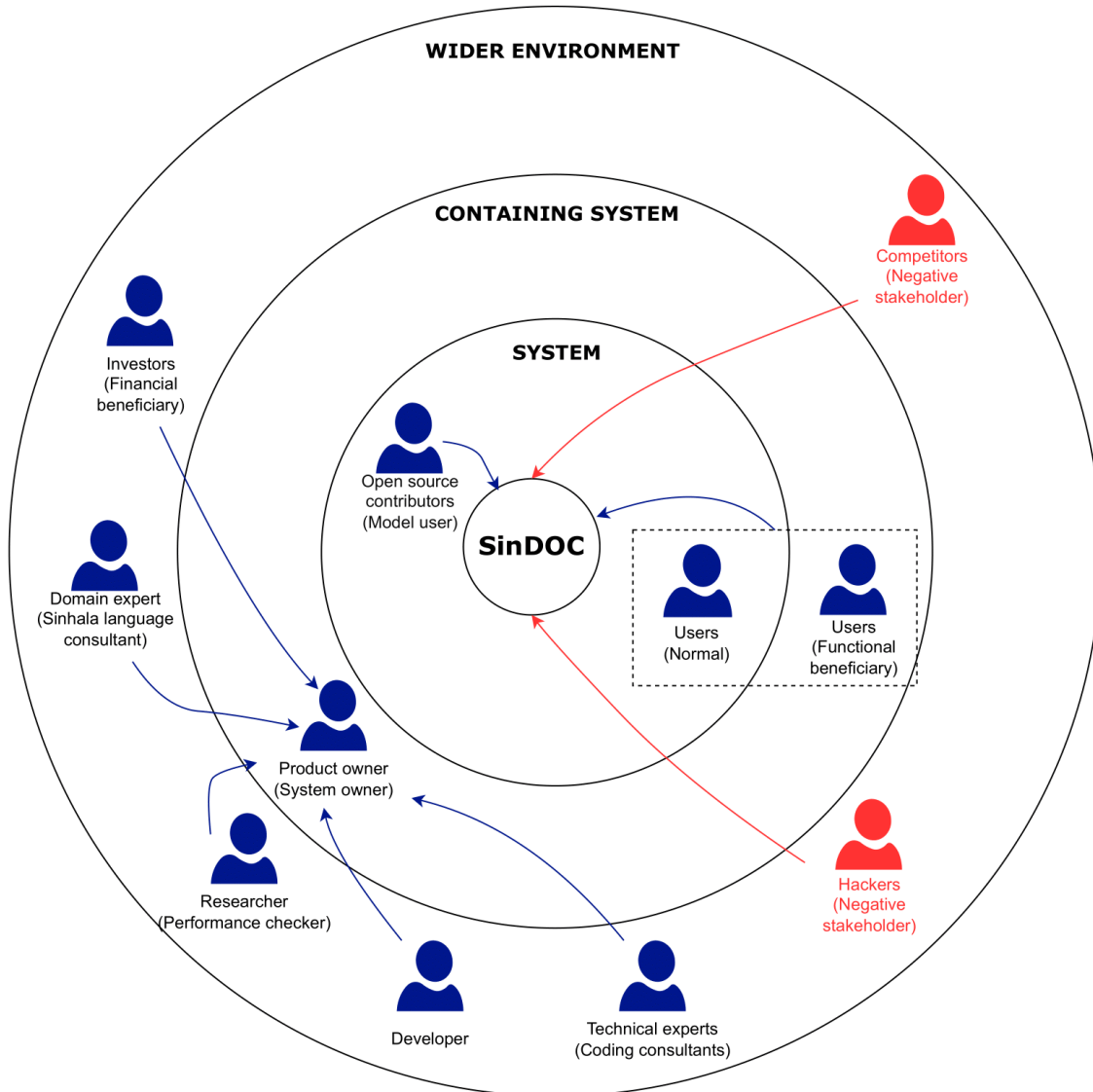


Figure 5: Stakeholder Onion Model

4.3.2. Analysis of the Stakeholder

Table 8: Stakeholder Analysis

Stakeholder	Role	Role description / Viewpoint
Inner System		
Users (Visitors)	Fundamental beneficiary	They are user who will be visiting the app for summarizing Sinhala documents
Open-source contributors		They will be using SinDOC’s open-source model for their own projects
The Containing System		
Users	Functional beneficiary	These users will be functionally benefited from the app to generate summaries for their content
Product owner	System owner/ operational / admin	Operates the staff and ensures if the product runs smoothly on available platform
Wider Environment		
Developers	Operational/ maintenance	They are responsible for developing the system using necessary tags and maintaining the quality of system by managing bugs and fixes
Technical experts	Experts/ Quality regulators / Consultants	Check for code quality and provides consultation on further improvements
Domain experts		Check for app functionality with the impact on the domain and provides consultation on further improvements
Researchers		Research on other existing models and literatures

		and provide consultations for app improvements.
Investors	Financial beneficiary	Provide necessary funds to improve the system and ensures to collect profits
Competitors	Negative stakeholders	Build a system that provides similar functions with better quality
Hackers		Illegally get into the system to access system data

4.4. Selection of Requirement Elicitation Methodology

To collect all necessary requirements the author followed several requirement elicitation methodologies. Literature review, survey, interviews, and brainstorming were the requirement elicitation methodologies used for gathering requirements related to this project.

Table 9: Requirement Elicitation Methodologies and Justifications

Method 1: Literature review
The author has done research on existing work related to the specific field. This helps in identifying the gaps that are up to be solved and their future enhancements. By studying them and comparing their systems the author was able to gather requirements using this technique
Method 2: Survey
A survey or a questionnaire was created to understand consumers or lead users' insights on an application that fulfills the aims of the research. This helps the author in understanding the overall customer experience they will be needing in an application that fulfills the goal.
Method 3: Interview
Formal interviews were conducted on both domain and technical experts. This helps in gathering expert knowledge and insights on going forward with the proposed solution. This also helps in identifying the gaps that are faced in the industry's on trying to overcome the problems discussed in the previous chapters. As an overall this method of requirement

gathering helps in identifying all challenges or drawbacks that will be faced when implementing the solution.

4.5. Discussion of Findings

4.5.1. Literature Review

Table 10: Literature Review Findings

Citation	Findings
(Dhananjaya et al., 2022)	The first approach to create a pre-trained model for Sinhala text classification
(Hasan et al., 2021)	Introduces a multilingual dataset for 44 low resource languages that are used for abstractive summarization
(Wolf et al., 2020)	A library created for transformer architecture that can be used for summarization tasks.
(Liu and Lapata, 2019)	Extractive and abstractive text summarization using pre-trained model BERT.
(Rathnasena et al., 2018)	Provides a method of computer vision and extractive summarization to extract text from old sinhala documents.

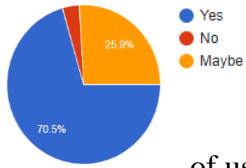
4.5.2. Survey

A survey was circulated around a group of Si Lankan individuals to identify the importance of this application within the community. The 112 responses and their findings are stated in the table below. The survey shared is attached in **APPENDIX D**.

Table 11: Survey Findings

Question	Are you a person who reads Sinhala documents?
Aim of the question	To understand the amount of people who read Sinhala documents

Findings



The results show that most of the users who participated tend to read Sinhala documents. Only less than 5% of the participants do not read Sinhala documents. It can be concluded that there is a potential number of users who will benefit from this application.

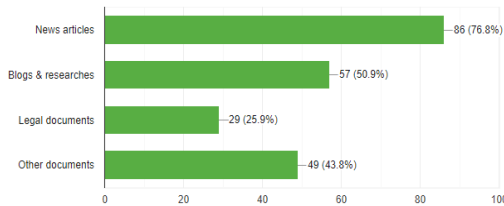
Question

What type of Sinhala documents do you read?

Aim of the question

To understand what type of Sinhala documents Sri Lankans read the most

Findings and conclusions



The results show that most of the participants read Sinhala news articles. Then blogs and research. This helps the author to understand in which domains the app could have a high impact.

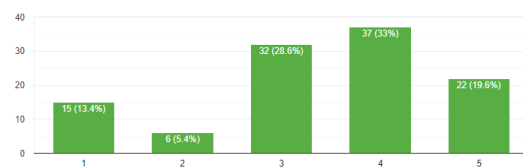
Question

How often do you read Sinhala documents?

Aim of the question

To understand the impact of Sinhala documents towards the community.

Findings and conclusions



These results show that most of the participants fall in the range of neutral (3), often (4) and very often (5) when it comes to their frequency of reading Sinhala documents. This shows that there is a good potential number of users who frequently read Sinhala documents.

Question

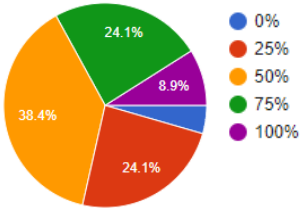
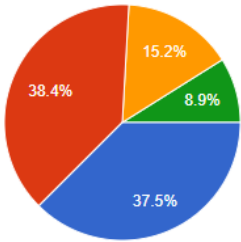
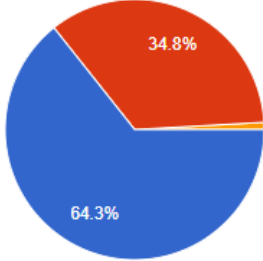
How much do you complete when reading such documents?

Aim of the question

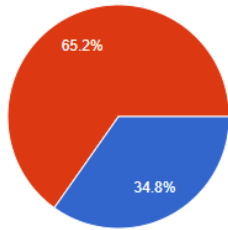
To identify if the users complete reading those documents.

Findings and conclusions

The results show only a few participants complete 100% of the document they read whereas

	<p>most of the participants read only half of the document. It can be concluded that most of the users miss important content of the document</p>
Question	<p>Do you get bored when reading long Sinhala documents?</p>
Aim of the question	<p>To understand the level of frustration faced by the community when reading such documents</p>
<div data-bbox="477 680 818 716"> Findings and conclusions </div> <div data-bbox="212 743 451 982">  </div> <div data-bbox="477 737 1411 995"> <p>The results show that around 76% of the participants fall into the range of getting bored all the time and very often. This shows that most of the participants do get frustrated. It can be concluded that the impact of this app can help in reducing the level of frustration.</p> </div>	
Question	<p>Would you like to get a summarized version of your document in Sinhala?</p>
Aim of the question	<p>To find the importance of creating this application that summarizes their Sinhala documents</p>
<div data-bbox="185 1318 532 1354"> Findings and conclusions </div> <div data-bbox="207 1423 467 1682">  </div> <div data-bbox="509 1373 1411 1577"> <p>The results show that almost 98% of the participants have the interest of trying an app that summarizes Sinhala documents. It can be concluded that the application the author is about to implement would benefit a large amount of people</p> </div>	
Question	<p>Have you used any Sinhala summarization tools before?</p>
Aim of the question	<p>To identify if users have used any other competitive tools</p>

Findings and conclusions



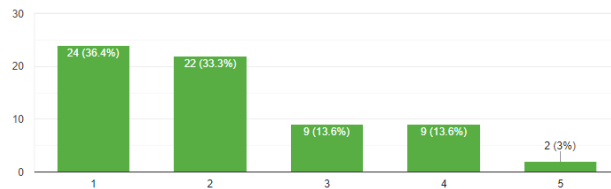
The results show that most of the users haven't come across a tool that summarizes Sinhala documents. The participants who have used an app have also had an average experience. The results will be discussed in the next finding. It can be concluded that most users haven't still come across a good Sinhala document summarizer

Question

If yes, how accurate do you think they are?

Aim of the question

If the users have used such tools this will help in identifying the performance of such tools and create a better application



Findings

Most of the participants who have used a Sinhala summarizer have stated that they have had a very bad or less experience with. It can be concluded that the users haven't come across with a better application until

now.

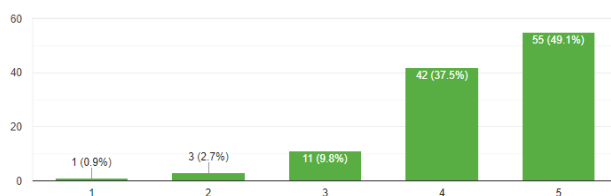
Question

How useful do you think a Sinhala document summarization tool would be for you

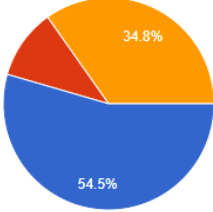
Aim of the question

To understand the level of impact this application could have within the Sri Lankan community.

Findings



Around 88% of the participants have stated that they might get great value on a Sinhala summarizer. This helps the author conclude that there is a high demand of this type of an application

Question	Would you prefer to have a model that summarizes many languages including Sinhala?
Aim of the question	To identify if the users would like additional features other than Sinhala summarization
<div style="display: flex; align-items: center;">  <div style="margin-left: 20px;"> <p>Findings</p> <p>This shows that only around 10% of the participants are looking for a Sinhala only summarizer. This helps the author identify that that it would be better to build an application with multi-language support.</p> </div> </div>	

4.5.3. Formal Interview

To conduct the interviews, the author decided to conduct interviews for technical experts. Two NLP experts, one data scientist, one senior campus lecturer and four data science students were interviewed. Their analysis and findings are stated in the table below. The proof of interviews is attached in **APPENDIX D**.

Table 12: Formal Interview Findings

Codes	Theme	Findings
<ul style="list-style-type: none"> • Sentence extraction • Use of existing datasets 	Dataset collection	<p>Experts advised gathering details from the community to identify which type of documents they read frequently and then find a dataset that matches the requirement.</p> <p>They suggested to use existing general datasets if there is no dataset related to users' requirements since the same development methods could be used for any Sinhala summarization task if trained accordingly.</p> <p>Use of google translate was highly</p>

		recommended if a new dataset would be created by the author.
<ul style="list-style-type: none"> • NLP techniques for Sinhala language • UTF-16 stopwords list preparation 	Data preparation and pre-processing	Identifying the right stopwords with UTF-16 could be a challenge but that will bring out the best results. So, they suggested creating such a list if there is not publicly available stopwords list.
<ul style="list-style-type: none"> • Dataset identification • Sinhala tokenizer identification 	Limitations of summarization techniques for low resource languages	Experts said that normal tokenizers will not work for Sinhala languages so identifying tokenizers for Sinhala language and using them in the model would be the best way. They also said to research some latest tokenizers that have been implemented for the Sinhala language.
<ul style="list-style-type: none"> • Use of pre-trained models • Combined models for Sinhala • Simple UI 	Model development	Using pre-trained models was highly recommended since they could save computation power and provide better results. They also suggested using both extractive and abstractive models for development. They advised to build a simple UI for the application so that the user understands the application features as soon as possible
<ul style="list-style-type: none"> • Qualitative and quantitative measures • Use multiple evaluation methods 	Evaluation techniques	Experts suggested using multiple summarization evaluation methods, as summarization testing is comparatively different to other NLP tasks, they said to not stick to one evaluation metric. They said that there is no perfect quantitative

		evaluation method for summarization, so it is also better to get evaluation from users after building the model.
--	--	--

4.6. Summary of Findings

Table 13: Summary of Requirement Findings

ID	Findings	Literature Review	Survey	Interview
1	Validation of research gap and problem statements	X	X	X
2	Identify impact on Sinhala domain	X	X	
3	Identify the existing works	X	X	
4	Identification of relevant dataset		X	X
5	Use of pre-trained models for gaining better result	X		X
6	Use of a combined approach for summarization techniques.	X		X
7	Provide option of extractive, combined and abstractive	X		X
8	Applicable for other languages also		X	
9	Simple User interface with a text input and output		X	X

4.7. Context Diagram

The context diagram provides insight of the system boundaries before implementing the solution. So, the following diagram provides an overview of the interaction between a user and the system whereas the user will have to enter a document's text and choose the type of summary and the system will provide the related summary of the document's content.

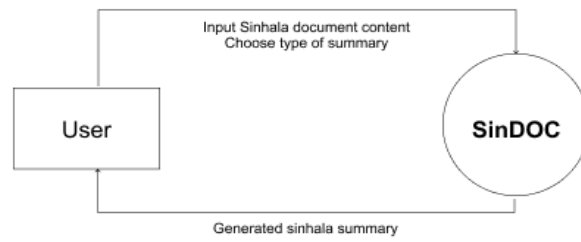


Figure 6: Context Diagram (Self-composed)

4.8. Use Case Diagram

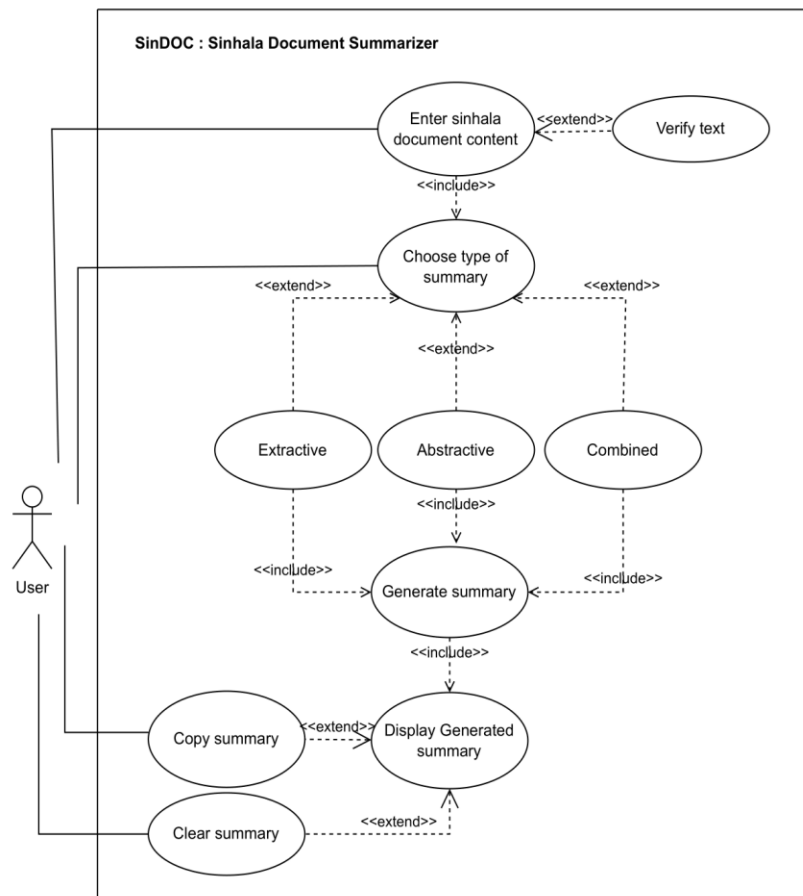


Figure 7: Use-case Diagram (Self-composed)

4.9. Use Case Description

The most important use case description is stated here. The other use cases are attached in the **APPENDIX D**.

Table 14: UC2 Use Case Description

Use Case Name	Choose type of summary
Use Case ID	UC2
Description	The user will have to choose the type of summary that needs to be used for the document summarization.
Priority	High
Actors	User
Pre-conditions	The user might want to have an understanding of the differences between the type of summaries
Extended use cases	Abstractive, Extractive. Combined
Included use cases	None
Main flow	<ol style="list-style-type: none">1. User chooses the type of summary.2. System checks if the type of summary is checked.3. If type of summary is checked it will send into model or an alert will be prompted4. Generates a summary of the Sinhala document inputted according to the selected type of summary.
Alternative flow	None
Exceptional flows	Alert if no choice is selected.
Post conditions	The user should be able to get a summary of the document according to their choice of summary

4.10. Requirements

To identify the requirements of the system and their prioritization levels, the 'MoSCoW' principle has been used. The elaboration of the principle is stated in the table below.

Table 15: MoSCoW Principles

M	Must have	Requirements that are a must or mandatory to be implemented. Cannot be ignored
S	Should have	Requirements that should be implemented. They cannot be necessary but still important
C	Could have	Requirements that could be implemented. They do not need to be implemented. It can be considered as a future improvement to the system
W	Will not have	Requirements that won't be implemented the system proposed

4.10.1. Functional requirements

The table below states the functional requirements that will be in the proposed system,

Table 16: Functional Requirements

FRID	Requirement description	Priority level	Use case
FR01	Option of inputting the document text into the system	M	Enter Sinhala document content
FR02	Option of choosing type of summary that needs to be generated	M	Choose type of summary
FR03	Verify text	M	Verify text
FR04	Generate extractive summary	M	Extractive
FR05	Generate abstractive summary	M	Abstractive
FR06	Generate combined summary with both extractive and	M	Combined

	abstractive models.		
FR07	Display the generated summary to user	M	Display generated summary
FR08	Alert if no choice of summary is selected	S	Choose type of summary
FR09	Copy the generate summary to clipboard	S	Copy summary
FR10	Clear the generated summary	S	Clear summary
FR11	Save all generated summaries to a database	C	Display generated summary
FR12	Multiple language summarization support	C	Generated summary
FR13	Input pdf documents	W	Enter Sinhala document content
FR14	Summarize images	W	Enter Sinhala document content

4.10.2. Non-functional requirements

The table below states the non-functional requirements that will be in the proposed system,

Table 17: Non- Functional Requirements

NFRID	Requirement description	Priority level	Requirement
NFR01	Implemented system needs to be efficient and	M	Performance

	effective in generating outputs as quickly as possible.		
NFR02	Implementation should be implemented by considering the user experience they will have within the application. The user interface of the application will support on improving the customer experience	M	Usability
NFR03	Maintaining users' privacy or data without breaching them should be considered when implementing the proposed system	S	Security
NFR04	Updating the system accordingly for a longer run could improve the number of consumers. For this the system need to be maintained accordingly and be updated according to latest industry principles	C	Maintainability
NFR05	When the number of consumers increases the ability to maintain multiple users without causing errors will be and important aspect. Scaling the system accordingly will reduce these types of errors and provide a better experience	C	Scalability

4.11. Chapter Summary

This chapter mainly describes the stakeholders and their impact within this research. The rich picture provides a helicopter view of the research concept, and the stakeholder onion model describes all stakeholders involved in the project and their impact. The author has defined the requirement gathering strategies taken. The context diagram shows a Level 0 data flow of the system. The use case diagram and descriptions highlight the most important functions that will be included within the system. And finally, the author has identified and stated the functional and non-functional requirements of the proposed solution.

CHAPTER 05: SOCIAL, LEGAL, ETHICAL AND PROFESSIONAL ISSUES

5.1. Chapter Overview

This chapter describes the social, ethical, legal and professional issues that the author should consider while working on this research.

5.2. Breakdown of Social, Legal, Ethical and Professional Issues

Table 18: SLEP Issues

Social	Legal
<ul style="list-style-type: none">• None of the participants' details were gathered when collecting responses for the survey.• The research does not include the name of any domain and technical related interviewees.• The information that has been gathered from third parties will be erased when the project has been completed	<ul style="list-style-type: none">• The dataset used for the project is publicly open for contributions.• Relevant dataset citations have been given on the relevant deployed platforms.• The pre-trained models used are also open source.• All development tools used are open source.
Ethical	Professional
<ul style="list-style-type: none">• Survey participants and expert interviewees were clearly briefed on how they will be contributing to this research.• Plagiarism and false details were highly considered and avoided with this research.	<ul style="list-style-type: none">• All surveys, interview responses and details are privately stored which are only accessible by the author.• Limitations, evaluations are stated as it is without faking them.

5.3. Chapter Summary

This chapter includes the SLEP issues that will be considered and followed by the author throughout the research to align with the BCS Code of Conduct.

CHAPTER 06: DESIGN

6.1. Chapter Overview

This chapter includes all design related decisions made by the author to identify the suitable methods; structures required for implementation. The goals and design diagrams are also stated in the following chapters.

6.2. Design Goals

Table 19: Design Goals

Performance	The system should run without any failure. Both front end and backend should smoothly interact with each other and provide results quickly. The trained model should have enough efficiency to provide quality outputs without consuming much time
Usability	Accessibility and usability of the system need to be considered. As the system will be available for any type of user the GUI should be as user friendly as possible. This will help the user to understand the system as quickly as possible.
Adaptability	Since this field of study is supposed to be an active trend for the foreseeable future the system has to have the ability to adapt accordingly. The model needs to be trained with better data sets and provide better results.
Reusability	Since this type of a project can benefit in other tasks within the domain it needs to be available for others to reuse the code and implement it in their own ways. For this case making the code open source and available for everyone will make it reusable

6.3. High Level Design

6.3.1. Tiered Architecture of the Proposed System

The following diagram depicts the three-tier architecture diagram that shows the architecture of the proposed system.

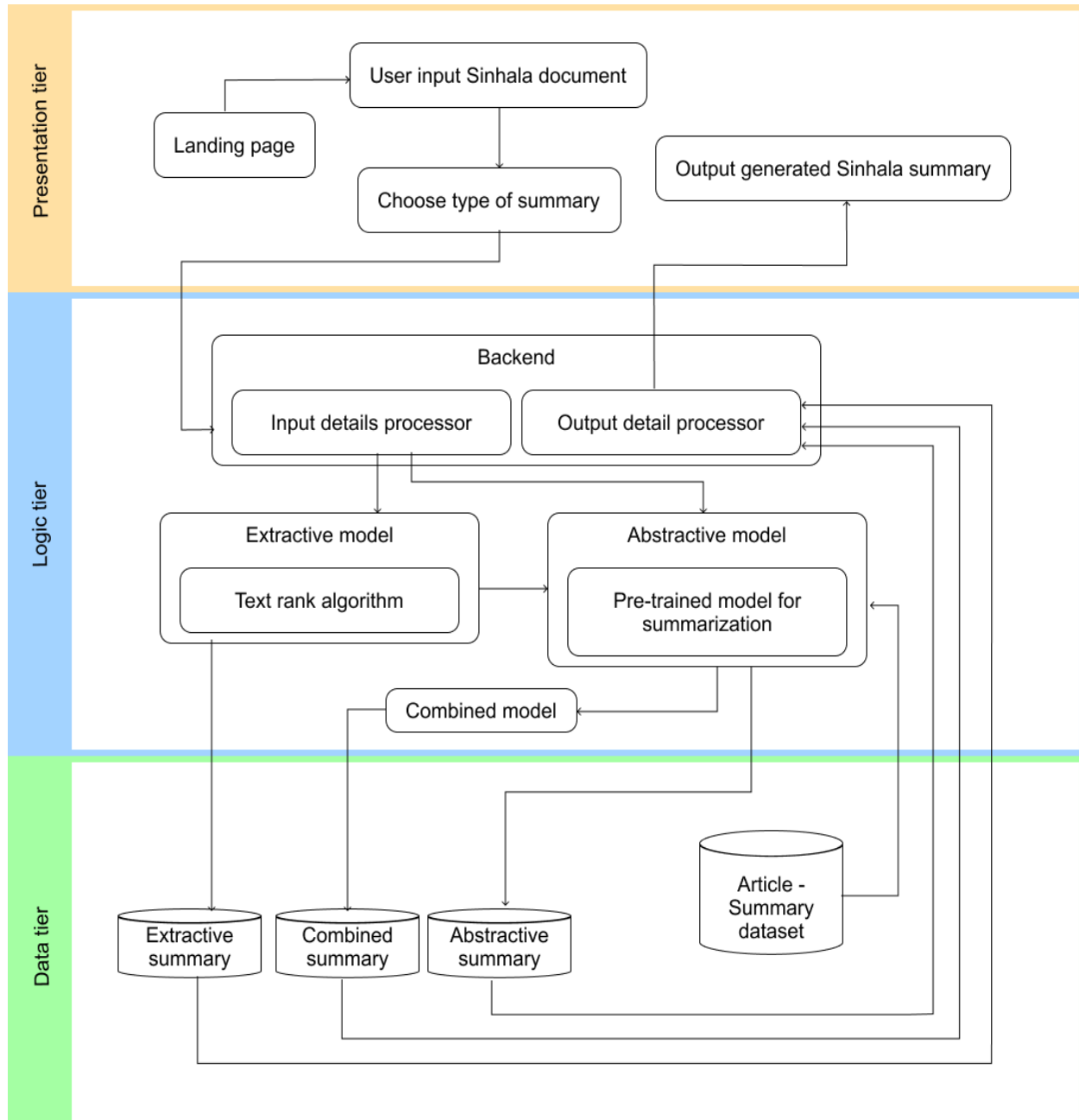


Figure 8: High-Level Architecture Diagram (Self-composed)

6.3.2. Discussion of tiers of the architecture

Data tier

1. Article-summary dataset - This includes the train, test and evaluate datasets that will be used to train the pre-trained model for abstractive summary generation.
2. Extractive summary - This will store the extractive summary generated by the extractive model and pass it to the user if that is his or her choice of summary.
3. Abstractive summary - This will store the abstractive summary generated by the abstractive model and pass it to the user if that is his or her choice of summary.
4. Combined summary - This will store the combined summary generated by the hybrid model and pass it to the user if that is his or her choice of summary.

Logic tier

1. Backend -
 - a. Input details processor - This will gather the input text and choice of summary chosen by the user in the client side and pass them to the respective model for document summarization tasks.
 - b. Output details processor - This will get the generated document summary and send it back to the client side so that the user will be able to use the generated summary.
2. Extractive model - This will use text rank algorithm to generate sentence scores and create summaries.
3. Abstractive model - This will be a model that will be transfer learned by a Sinhala article summary dataset to generate summaries.
4. Combined model - This will be a model that will use extractive summarization first and then pass it to the abstractive model and generate summaries.

Presentation tier

1. Landing page - This includes the navigation to the app and the functionality descriptions of the application.
2. Input Sinhala document - This will be the field where the user will have to enter the document text in which he or she would like to summarize.
3. Choose type of summary - This will be the option where the user will have to choose what type of summary he or she would like to get as a result.
4. Output generated summary- This will show the result of the generated summary to the user.

6.4. System Design

The author identified two of the most widely used system design paradigms. They are namely **OOAD** (Object Oriented Analysis and Design) and **SSADM** (Structured System analysis and Design Method). The design paradigm and the reasoning are stated in the following chapter.

6.4.1. Selection of design paradigm

After researching the above stated design paradigms, the author has decided to follow the SSADM design paradigm for the implementation of this system. The reasons are stated as follows.

- Since the project is mainly based on a data science component, usage of object-oriented principles will not be that beneficial.
- MVP structure can easily be followed and implemented by following this design paradigm.
- The programming languages that are used in this system don't support OOP by nature.

6.5. Design Diagrams

6.5.1. Data flow diagram

The Level 0 or context diagram-based data flow diagram can be found in the **Chapter 4.7**. The Level 01 DFD and Level 02 DFD can be seen below. The Level 01 DFD provides a basic view of the SinDOC app functionality and the Level 2 DFD gives an elaborated version of the Level 01 DFD.

Level 01 Data Flow Diagram

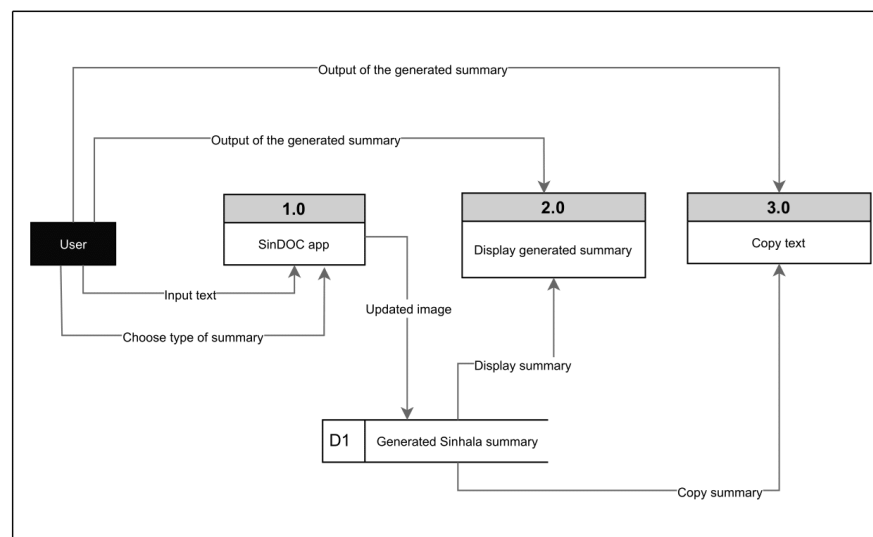


Figure 9 : Level 01 DFD (Self-composed)

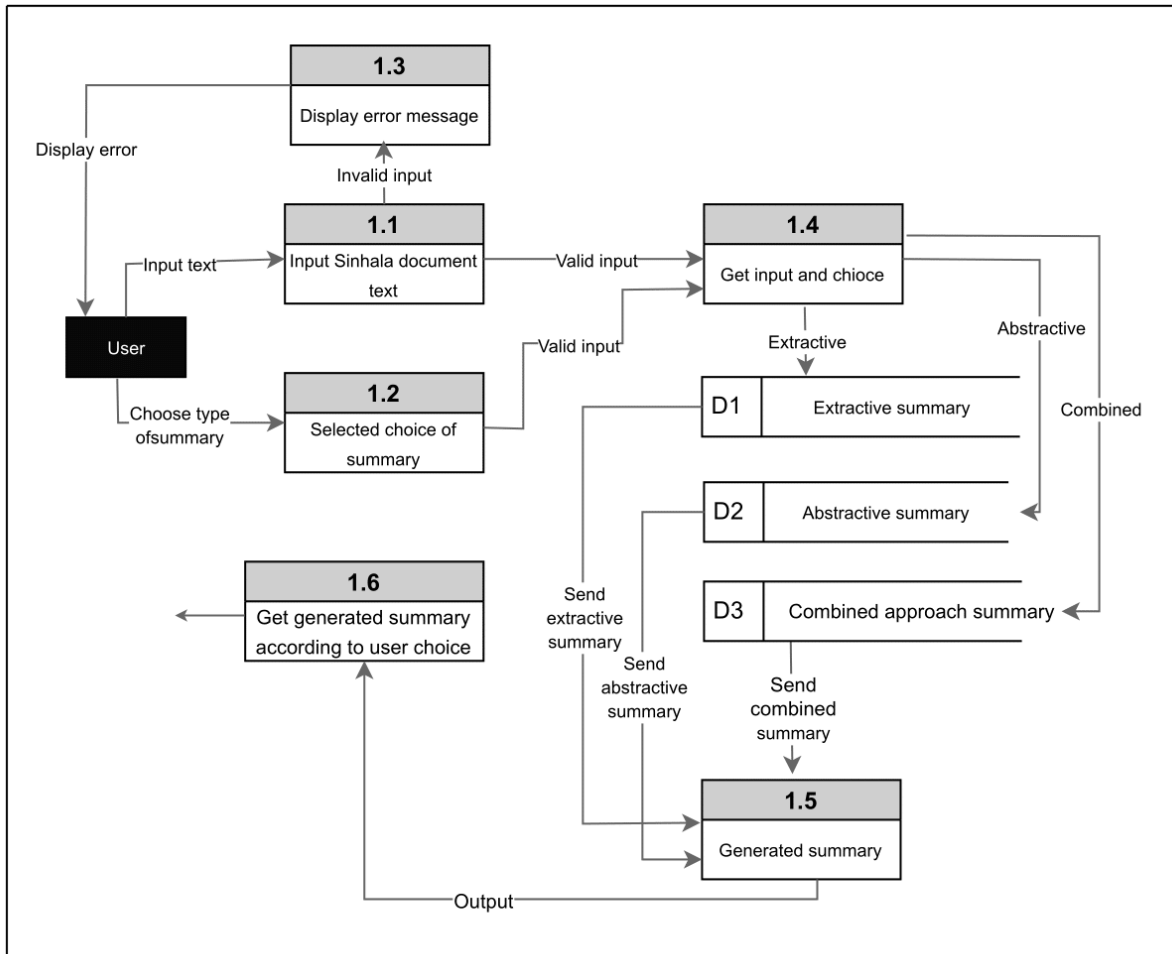


Figure 10: Level 02 DFD (Self-composed)

Level 02 Data Flow Diagram

6.5.2. UI design

When designing the UI, user experience was highly considered. User research was conducted prior to the designing phase. The user research helped the author in identifying the type of application that needs to be designed. Finally, the author decided to go with a minimalistic light theme application that can be used by any individual within any age group. The low and highly fidelity designs are attached in the **APPENDIX E**.

6.5.3. System Process Flowchart

The following diagram shows the users activity pathway within the application. The process contains steps from inputting a long Sinhala document to getting a summarized version of it.

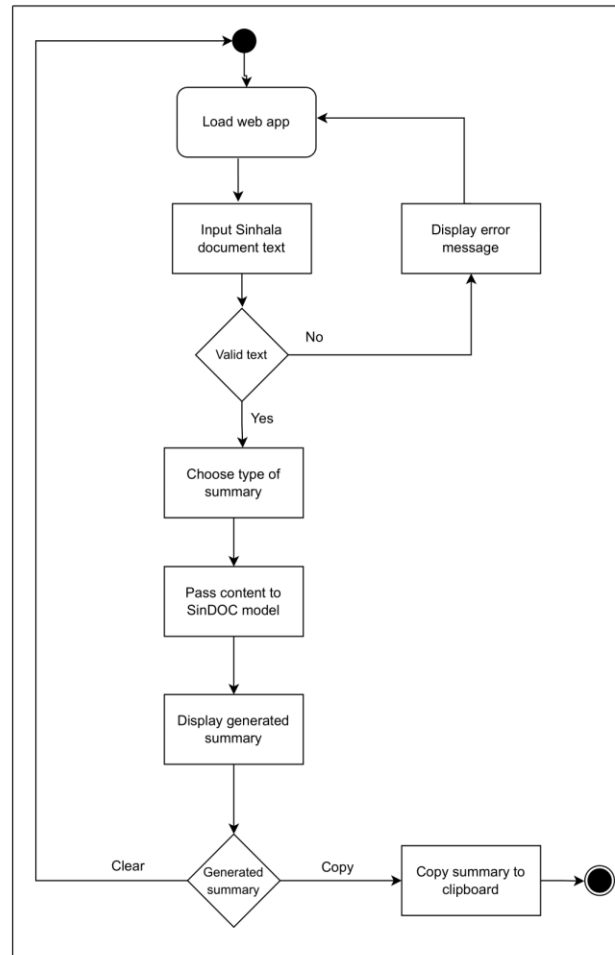


Figure 11: System Process Flowchart (Self-composed)

6.5. Chapter Summary

This chapter contains details on how the system designs will be implemented. The design goals are initially stated. The tiered diagram shows how the system will be functioning on each tier. The selection of the system design and its reasoning are then stated. The DFD diagrams for Level 1 and 2 are then composed to provide a deep view of the system data flow. The UI Design that will be followed to design the front end is also included. The system process flowchart or the activity diagram that a user would follow is finally mentioned in this chapter.

CHAPTER 07: IMPLEMENTATION

7.1. Chapter Overview

This chapter includes all core implementation related details and code. The use of programming languages, IDE's, libraries, and dataset are all discussed and summarized within this chapter.

7.2. Technology Selection

7.2.1. Technology Stack

The technology stacks used in the presentation, logic and data tiers are stated in the following image.

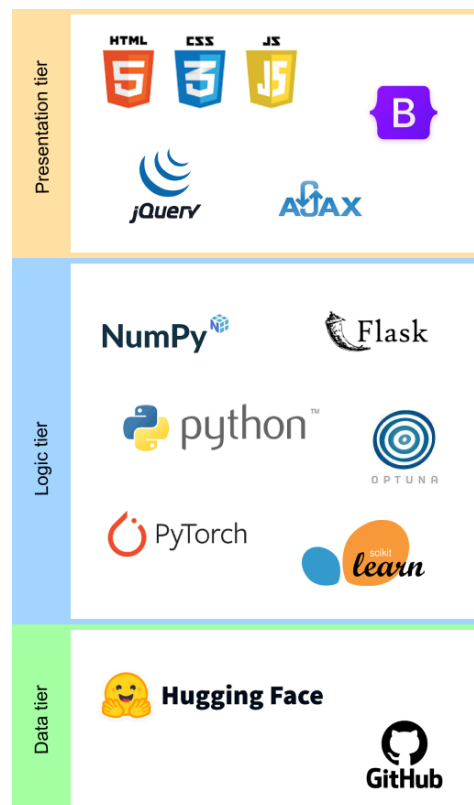


Figure 12: Technology Stack

7.2.2. Dataset Selection

There are many publicly available datasets for document and text summarizations tasks. Some highly recommended and used datasets are stated in the table below.

- WikiSummary - Articles and their summaries generated from Wikipedia.

- Multi-X-Science - Contains multi-document summaries of research articles.
- Gigaword - Headline generation dataset for around four million articles
- BigPatent - US patent documents with human summaries
- BBC News Summary - Dataset of BBC news articles and their summaries. Includes 44 separate language datasets.
- CNN Daily mail - Contains almost 400k CNN news articles and their human generated summaries.

The author decides to select the ‘**CNN / DailyMail Dataset**’ for this project. As it has unique summaries of news articles generated by humans. The author managed to translate the dataset to relevant language to fill the research gap that was identified. A new sub dataset is created which can be denoted as a dataset contribution.

7.2.3. Development Frameworks

In order to maintain a reliable and efficient code to witness the research performed by the author, simple frameworks were chosen. Flask was chosen as the python framework to run the backend of the project. Bootstrap was used to build the front end of the project. The author describes the chosen framework with necessary justifications.

Table 20: Development Frameworks

Flask	Flask highly supports python libraries which will help the author to build the web application
Bootstrap	This framework helps in creating responsive UI elements with ease. The author decided to use this framework to build the components of the web apps frontend

7.2.4. Programming Languages

The author chose to use **Python** as the desired programming language while comparing it to similar other languages like R and Java. Since python is the best programming language for data science related work, the author used it extensively for the development of the project. The backend of the project is carried out in python which makes it easier to handle the data science components along with it. The capability of having multiple libraries that support the author's

research component has also been a reason for the choice of python as the programming language.

7.2.5. Libraries

The libraries and their justifications are stated in the table below.

Table 21: Libraries used for Implementation

Hugging Face	Includes high quality pre-trained models which will be studied and used in this research
Pytorch	This is a widely used library for NLP tasks. This will be used for summarization tasks
Optuna	A framework used for optimizing hyperparameters to identify the best ones for the model.
NLTK	It provides resources for tokenization, text processing, stemming and similar NLP tasks
Numpy	Mathematical functions are implemented using this library
Pandas	This library provides the flexibility of cleaning, preprocessing and preparing the data according to the research need

7.2.6. IDE's

The IDE's and their justifications are stated below.

Table 22 : IDE's used for Implementation

Google Colab Pro	The use of NVIDIA P100, Tesla T4 and A100 GPU models has helped a lot in training the model that achieves the best result
VS Code	The robust and extensible architecture of this helps the author to build the web application with ease.

7.2.7. Summary of Technology Selection

Table 23: Tools and technology Stacks

Component	Tool or Technology
Programing Language	Python
Development Framework	Flask
UI Framework	Bootstrap
Libraries	Numpy, Pytorch, NLTK, Optuna
IDE – Model	Google Collab Pro
IDE – SinDOC prototype	Visual Studio Code
Version Control	Github, Huggingface

7.3. Implementing of the core functionality

Building a combined approach for summarization which involves both extractive and abstractive techniques has been the goal of this research project. For this the author had to initially identify a valid dataset and prepare it for model training. Once the dataset has been identified the author went on building the extractive model which uses word frequency and sentence scoring algorithms to extract the most important sentences of a document. Once the extractive model extracts the most important sentences it is then sent into the abstractive model for paraphrasing and restructuring it as a human generated summary. This complete process is called the combined approach in which both summarization techniques are used for summarization. Through the implementation the author was able to get the best of both models by extracting keywords (extractive) and restructuring it with own words (abstractive) to generate almost human-like summaries.

7.3.1. Extractive Model Implementation

Initially a stopwords was found and used for stopwords removal.

```
a=[]
with open('/content/drive/MyDrive/FYP - final/stopWords.txt', 'r',encoding="utf-16") as f:
    a+=f.readlines()
f.close()
for i in range(0,len(a)):
    a[i]=a[i].rstrip('\n')
stopWords = a
```

Figure 13: Stopword Removal (Extractive)

Generate frequency table for words.

```
def _create_frequency_table(text_string) -> dict:

    words = word_tokenize(text_string)
    ps = PorterStemmer()

    freqTable = dict()
    for word in words:
        word = ps.stem(word)
        if word in stopWords:
            continue
        if word in freqTable:
            freqTable[word] += 1
        else:
            freqTable[word] = 1

    return freqTable
```

Figure 14: Word Frequency Table Generation (Extractive)

Scoring sentences using the word frequency results and finding average score

```
def _score_sentences(sentences, freqTable) -> dict:
    sentenceValue = dict()

    for sentence in sentences:
        word_count_in_sentence = (len(word_tokenize(sentence)))
        word_count_in_sentence_except_stop_words = 0
        for wordValue in freqTable:
            if wordValue in sentence.lower():
                word_count_in_sentence_except_stop_words += 1
            if sentence in sentenceValue:
                sentenceValue[sentence] += freqTable[wordValue]
            else:
                sentenceValue[sentence] = freqTable[wordValue]

        if sentence in sentenceValue:
            sentenceValue[sentence] = sentenceValue[sentence] / word_count_in_sentence_except_stop_words

    print(sentenceValue)
    return sentenceValue

def _find_average_score(sentenceValue) -> int:
    sumValues = 0
    for entry in sentenceValue:
        sumValues += sentenceValue[entry]

    # Average value of a sentence from original text
    average = (sumValues / len(sentenceValue))

    return average
```

Figure 15: Sentence Score Generation (Extractive)

Generating summary by finding the threshold of the average of the sentence scores

```
def _generate_summary(sentences, sentenceValue, threshold):
    sentence_count = 0
    summary = ''

    for sentence in sentences:
        if sentence in sentenceValue and sentenceValue[sentence] >= (threshold):
            summary += " " + sentence
            sentence_count += 1

    return summary
```

Figure 16 : Threshold Identification Using Average Sentence Scores (Extractive)

Implementing the extractive summarizer that passes the text to all functions above.

```
def extractive_summarizer(text):  
    # Create the word frequency table  
    freq_table = _create_frequency_table(text)  
  
    # Tokenize the sentences  
    sentences = sent_tokenize(text)  
  
    # Score the sentences  
    sentence_scores = _score_sentences(sentences, freq_table)  
  
    # Find the threshold  
    threshold = _find_average_score(sentence_scores)  
  
    # Generate the summary  
    summary = _generate_summary(sentences, sentence_scores, 1.0 * threshold)  
  
    return summary
```

Figure 17: Extractive Summarizer Function (Extractive)

7.3.2. Dataset Creation

In order to train the abstractive model a dataset was created. A subset of the CNN daily mail dataset was translated to create Sinhala articles and summaries. The CNN daily mail dataset includes an approximate of 240000, 14000 and 12000 rows for train, validation, and test respectively. A subset was created where 6000, 2000 and 2000 rows were gathered from the train, validation and test datasets and then translated. The following image shows the first five rows of the CNN daily mail.

	article	highlights	id
0	LONDON, England (Reuters) -- Harry Potter star...	Harry Potter star Daniel Radcliffe gets £20M f...	42c027e4ff9730fbb3de84c1af0d2c506e41c3e4
1	Editor's note: In our Behind the Scenes series...	Mentally ill inmates in Miami are housed on th...	ee8871b15c5d0db17b0179a6d2beab35065f1e9
2	MINNEAPOLIS, Minnesota (CNN) -- Drivers who we...	NEW: "I thought I was going to die," driver sa...	06352019a19ae31e527f37f7571c6dd7f0c5da37
3	WASHINGTON (CNN) -- Doctors removed five small...	Five small polyps found during procedure; "non...	24521a2abb2e1f5e34e682a0fe9e56904a2b0e88
4	(CNN) -- The National Football League has ind...	NEW: NFL chief, Atlanta Falcons owner critical...	7fe70cc8b12fab2d0a258fababf7d9c6b5e1262a

Figure 18: CNN Dataset before Translation

For translation of the dataset Google translate API was used. It is to be noted that google translates a maximum of 5000 characters at a time. So, to solve this problem the author skipped rows that had rows more than 4500 characters. The following image shows the implementation of dataset translation using google API.

```
count = 0
for index, row in df_filtered.iterrows():
    if len(row['highlights']) <= 4500:
        df_filtered.at[index, 'summary_sinhala'] = GoogleTranslator(source='auto', target='si').translate(row['highlights'])
    else:
        print(f"Skipping row {index} for 'highlights' column because the length exceeds 5000")

    if len(row['article']) <= 4500:
        df_filtered.at[index, 'article_sinhala'] = GoogleTranslator(source='auto', target='si').translate(row['article'])
    else:
        print(f"Skipping row {index} for 'article' column because the length exceeds 5000")

count += 1
print(count)
```

Figure 19:: CNN Dataset Translation

The following image shows the first five rows of the translated dataset

df_filtered					
	article	highlights	id	summary_sinhala	article_sinhala
0	LONDON, England (Reuters) – Harry Potter star...	Harry Potter star Daniel Radcliffe gets £20M t... 42c027e4ff750b36e4c1af02d506641c3e4	හාරි පොටර් තරුට වැඩිමයෙද් දික්වැට් සඳහා 18 ට...	ලන්ඩන්, එංගලන්තය (රෙයිටර්) – හාරි පොටර් තරුට...	
1	Editor's note: In our Behind the Scenes series...	Mentally ill inmates in Miami are housed on th... ee8871b15c500d07b017b01472ba6b350691f9	මියැපී මි මානසික ආබාධිත දැඩ්වැට් "අමතක වූ නිකුත්"	සාර්තානේ සබහාන: අනෙක් Behind the Scenes මාලාවේ C...	
2	MINNEAPOLIS, Minnesota (CNN) – Drivers who we...	NEW: "I thought I was going to die," driver sa... 06352019a19ac31e52737f751f6c0d70a50d37	අනුක් "මම නිකුත් මම මරණය වී සිටියා," රියදුරු පව්...	මිනියාපොලිස්, මිනසොටා (මිනර්වර්) – මිනියාපොලි...	
3	WASHINGTON (CNN) – Doctors removed five small...	Five small polyps found during procedure; non... 24521a2abb2e1f5e34e6824e0f9e569042be08	සරියා පටියානු අනතුරු කුඩා ඖෂධීය පොත් හමු වී...	මොෂිංට් (මිනර්වර්) – පොදුසුවරු රිකර්ට් පොහො...	
5	BAGHDAD, Iraq (CNN) – Dressed in a Superman	Parents beam with pride, can't stop from smili... a1ebbb6b44370a1ff26759206d572b5e0642d70	දෙමව්පියන් ආබාධිතයන් දිවිපොහොත්, සහපොහොත්...	බැග්ඩාඩ්, ඉරාක (මිනර්වර්) – ඉස්ලාමික සම්ප්පා...	

Figure 20: CNN Dataset after Translation

The created dataset has been added to the hugging face library. This helps future developers use it for their research and make contributions. The following link navigates to the dataset.

Hamza-Ziyad/CNN-Daily-Mail-Sinhala

7.3.3. Data pre-processing

Before using the dataset to train the model the dataset was pre-processed. The following steps were taken. All screenshots are attached in the **APPENDIX F**.

- Removal unwanted columns
- Removal of rows that has null values.
- Removal of HTML tags
- Removal of the word CNN in both English and Sinhala
- Removal of URL links
- Removal of two consecutive duplicate words

7.3.4. Abstractive Model Implementation

Initiating the large language model that will be used for fine-tuning. mT5 (Xue et al., 2021) was the model used in this case.

Using Optuna (Akiba et al., 2019) identifying the best parameters for training the model.

```
def objective(trial: optuna.Trial):
    model_name = model_checkpoint.split("/")[-1]

    args = Seq2SeqTrainingArguments(
        output_dir=f"sinMT5-hyper-tuned",
        evaluation_strategy="epoch",
        save_strategy="epoch",
        greater_is_better=True,
        predict_with_generate=True,
        load_best_model_at_end=True,
        weight_decay=trial.suggest_float("weight_decay", 4e-5, 0.01, log=True),
        learning_rate=trial.suggest_float("learning_rate", 4e-5, 0.01, log=True),
        num_train_epochs=trial.suggest_int("num_train_epochs", 4, 8),
        warmup_ratio=trial.suggest_float("warmup_ratio", 0.0, 0.1),
        per_device_eval_batch_size=trial.suggest_int("per_device_eval_batch_size", 4, 8),
        per_device_train_batch_size=trial.suggest_int("per_device_train_batch_size", 4, 8),
        save_total_limit=1,
    )

    #preparing model trainer
    trainer = Seq2SeqTrainer(
        model,
        args,
        train_dataset=tokenized_datasets,
        eval_dataset=tokenized_datasets_eval,
        data_collator=data_collator,
        tokenizer=tokenizer,
    )
```

Figure 21: Hyper Parameter Tuning (Abstractive)

Prepare the training arguments

```
logging_steps = len(tokenized_datasets) // 4
model_name = model_checkpoint.split("/")[-1]

args = Seq2SeqTrainingArguments(
    output_dir=f"sinMT5-tuned",
    evaluation_strategy="epoch",
    learning_rate=0.00015652249866150822,
    per_device_train_batch_size=4,
    per_device_eval_batch_size=4,
    weight_decay=9.511698790218645e-05,
    save_total_limit=3,
    num_train_epochs=7,
    predict_with_generate=True,
    logging_steps=logging_steps,
    push_to_hub=True,
```

Figure 22: Training Arguments (Abstractive)

Push the trained model to a hugging face. The model is made public for future research. The following link navigates to the model.

[Hamza-Ziyad/sinMT5-tuned](#)

And finally define the trained abstractive model in the combined approach. The implementation is in **APPENDIX F**.

7.3.5. Combine Approach Model Implementation

Finally, the combined approach was built where the input is initially passed into the extractive model and then passed into the abstractive model.

```
def combined_summarizer(text):
    extractive_summary = extractive_summarizer(text_str)
    print('Extractive Summary --> ',extractive_summary)

    abstractive_summary = abstractive_summarizer(text_str)
    print(' Abstractive Summary --> ',abstractive_summary)

    combined_summary = abstractive_summarizer(extractive_summary)
    print(' Combined Summary --> ',combined_summary)

    return combined_summary , extractive_summary , abstractive_summary
```

Figure 23: Combined Approach Function (Combined)

7.5. UI Implementation

The frontend of the application was built using the bootstrap framework and the frontend minimal functions were built through jQuery. The following two screens are the homepage and the output page. The rest of the screenshots can be found in the **APPENDIX E**.



Figure 24 : SinDOC Home (UI Implementation)

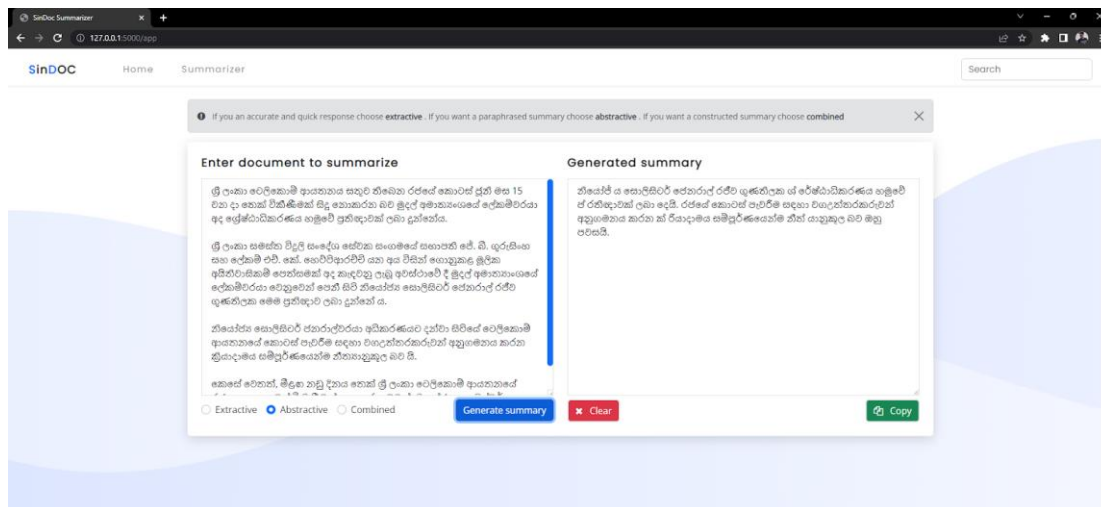


Figure 25: SinDOC Summarizer (UI Implementation)

7.4. Chapter Summary

This chapter talks about the core implementation and what technology stacks have been used for the implementation. Initially the technology stack has been discussed. The development frameworks, libraries and IDE's that will be used are also discussed. When it comes to the core

implementation the author has clearly stated the steps followed for implementation by attaching screenshots of necessary steps. The UI implementation methods and screenshots are also discussed in the final section of this chapter.

CHAPTER 08: TESTING

8.1. Chapter Overview

This chapter will be focusing on the testing methods carried out by the author to make sure that all the requirements have been met and the model results provide high quality when compared to other SOTA models. The objectives and goals of the whole testing process are also elaborated in the following sections.

8.2. Objectives and Goals of Testing

The ultimate goal of this research is to build an app that meets the research aim. The objective and goals are broken down into certain categories which are stated below.

- To make sure of the quality of built models and if it generates results as expected.
- Make sure if they Must and Should have options are met in the MoSCoW principle.
- What are possible future evaluation methods that can be used to test the performance of the application.
- Ensure that the most important functional and non-functional requirements are addressed.
- State the possible fixes that can be implemented in the future for this research solution.

8.3. Testing Criteria

The two most important testing criteria were considered for testing this research solution. The following were the testing criteria followed by the author.

- Functional testing - All functional requirements and their achievements come under this criterion.
- Structural testing - All non-functional requirements and their achievements come under this criterion.

8.4. Model Testing

8.7.1. sinMT5 Comparisons Before and After Hyper Parameter Tuning

The abstractive model named sinMT5 was tested on literature review suggested training arguments and as well as automated hyper parameters suggesting training arguments.

The model made using SOTA (Hasan et al., 2021) training results are stated below. Click this [link](#) to explore more details.

Training Loss	Epoch	Step	Validation Loss	Rouge1	Rouge2	RougeL
1.8746	1.0	750	1.8262	18.9753	7.9271	18.1349
1.4727	2.0	1500	1.8094	19.2219	7.9749	18.4314
1.2331	3.0	2250	1.8432	20.436	7.8378	19.584
1.0381	4.0	3000	1.8987	20.2251	7.9593	19.1556
0.8737	5.0	3750	1.9471	20.3262	7.8935	19.407
0.7363	6.0	4500	2.0611	20.1551	7.5046	19.2213
0.6214	7.0	5250	2.1838	19.9045	7.6232	18.743
0.5277	8.0	6000	2.3190	20.8581	8.1054	19.8079
0.4576	9.0	6750	2.4091	20.028	7.7635	19.0721
0.4099	10.0	7500	2.4863	19.9769	8.04	19.0307

Figure 26: Training Results without Hyper Parameter Optimization

The model uses automated hyper parameters tuning and getting the best training arguments training results are stated below. Click this [link](#) to explore more details.

Training Loss	Epoch	Step	Validation Loss	Rouge1	Rouge2	RougeL
1.8651	1.0	1500	1.8070	17.676	7.1418	16.8638
1.5527	2.0	3000	1.7804	21.1357	8.1386	20.122
1.3755	3.0	4500	1.7769	21.4151	8.5692	20.3204
1.2473	4.0	6000	1.7937	21.2434	8.2325	20.1332
1.1548	5.0	7500	1.8035	20.4298	8.2314	19.5909
1.0835	6.0	9000	1.8367	20.5427	8.2226	19.6134
1.0387	7.0	10500	1.8573	20.2531	8.1307	19.3917

Figure 27 Training Results with Hyper Parameter Optimization

It is clearly visible that the automated hyperparameter tuning model provides better performance with a smaller number of epochs. While the other model tends to go for more epoch and a highly increasing validation loss that makes the model overfitting. The results would state that the author's research could be a strong technical contribution that introduces hyper parameter tuning

for abstractive summarization. The parameters generated by the model are shown in **APPENDIX VI**.

8.7.1. Extractive, Abstractive and Combined Summarization Model Results

The average **ROUGE1/ROUGE2/ROUGEL** scores to their two decimal points generated by all three models on ten samples of test data from both the BBC dataset and CNN daily mail translated data are as follows. All the ten ROUGE scores screenshots are attached in the **APPENDIX G**.

Precision scores - p

Table 24: Precision values of Test Data Sample

	BBC dataset	CNN DailyMail Dataset
Extractive model	0.20/ 0.13/ 0.14	0.35/0.20/ 0.30
Abstractive model	0.46/ 0.21/ 0.26	0.69/0.37/ 0.43
Combined model	0.55/ 0.29 / 0.27	0.75 / 0.45/ 0.61

Recall scores - r

Table 25: Recall values of Test Data Sample

	BBC dataset	CNN DailyMail Dataset
Extractive model	0.95/ 0.61 / 0.69	0.87/ 0.59 / 0.70
Abstractive model	0.87 / 0.45 / 0.52	0.80 / 0.42 / 0.53
Combined model	0.86/ 0.38 / 0.45	0.70 / 0.35/ 0.49

F1 scores - f

Table 26: F1 values of Test Data Sample

	BBC dataset	CNN DailyMail Dataset
Extractive model	0.34/ 0.21/ 0.22	0.48/0.36/ 0.40
Abstractive model	0.59 / 0.26 / 0.36	0.75/ 0.38 / 0.50
Combined model	0.60 / 0.27 / 0.34	0.71/ 0.37/ 0.49

Results may be inaccurate in certain cases since ROUGE isn't highly optimized for low resource languages (Hasan et al., 2021)

It makes it hard to conclude on what model does better as they produce variations on all aspects. But all scores are decent as they align near 1.0 which means that the generated summaries do include many connections between the real summaries. Inorder to get the best testing results human evaluation is suggested by much previous research (Hsu et al., 2018). So, responses from human evaluators are stated in Chapter 9

8.5. Benchmarking

The test data is from a previous abstractive model that has been tested using the XLSum dataset and this model results (Hasan et al., 2021). The model used a BBC summary dataset for testing. The same summary test data was used to test the results. The results of the F1 scores are stated in the table below.

Table 27: Test Data Benchmarking Results

	multilingual XLSUM	Ours - Abstractive (sinMT5-tuned)	Ours - Hybrid
ROUGE 1	27.2901	58.7512	60.2513
ROUGE 2	13.3815	26.3214	27.3521
ROUGE L	23.4699	35.5231	34.2135

It is important to note that the SOTA model used for benchmarking hasn't stated their method of testing. They have just provided the score stating that it is from their test data. Due to time constraints the author picked ten random summaries from the test dataset and generated the above-mentioned ROUGE scores. Due to this the results provided above may not be fully accurate and reliable.

Hasan et al. (2021) also states that the results generated for low resource languages aren't really very accurate since it hasn't been highly optimized for low resource languages.

8.6. Functional Testing

All functions that are supposed to be within the application as stated in the software specification chapter are tested and stated in the following table.

Table 28: Functional Testing Results

FRID	Use case	Expected outcome	Actual outcome	Status
FR01	Enter Sinhala document content	User should be able to input the document content within the text area	User can input the document content within the text area	Done
FR02	Choose type of summary	Users should be able to pick one of the summary types out of extractive, abstractive or combined.	Users can pick one of the summary types out of extractive, abstractive or combined.	Done
FR03	Verify text	A prompt should be alerted if the text includes only one sentence or no sentence	A prompt is alerted if the text includes only one sentence or no sentence	Done
FR04	Generate extractive summary	Get the extractive summary generated within the output text area	Generated extractive summary is displayed.	Done
FR05	Generate abstractive summary	Get the abstractive summary generated within the output text area	Generated abstractive summary is displayed.	Done
FR06	Generate combined summary.	Get the combined summary generated within the output text area	Generated combined summary is displayed.	Done
FR07	Display generated summary	Display the relevant summary result to the user.	Generated relevant summary is displayed.	Done

FR08	Choose type of summary	A prompt should be alerted if no choice of summary is selected.	A prompt is alerted to ask for choosing a type of summary	Done
FR09	Copy summary	Displayed summary should be able to be copied to user's clipboard	Display summary is copied to clipboard	Done
FR10	Clear summary	Displayed summary should be cleared by the user	Display summary is cleared	Done
FR12	Generated multilingual summary	User could be able to input some of other high level language documents too (Eg: English)	Display summary according to input language	Done

8.7. Module and Integration Testing

The testing on modules and integrations and their statuses for this research project are stated in the following table.

Table 29: Module and Integration Testing Results

Module	Input	Expected outcome	Actual outcome	Status
Document content input	Enter document content	Paste Sinhala document content within the text area	Can paste Sinhala document content within the text area	Done
	Verify document content	Alert if no more than one sentence content is added to the text area	Alert prompted asking for longer document	Done
Choice of	Choose	Select one of the types	Can choose an option of	Done

summary selection	extractive, abstractive, or combined approach	of summaries to generate a summary	summary to be generated.	
	Verify if choice is selected	Alert if no choice selected	Alert prompted asking for a choice to be selected.	Done
Sinhala summarization model	Generate extractive, abstractive or combined summary according to user input.	Display generated summary	Displays generated summary	Done
Additional options	Copy generated summary	Copy summary to clipboard	Summary can be copied to clipboard	Done
	Clear generated summary	Clear generated summary	Summary can be cleared	Done

8.7. Non-Functional Testing

8.7.1. Performance

The web application showed a level of 74% for performance according to the Google lighthouse report. The screenshots of that and the CPU and memory usages are also attached below. These results to show that the web application performs well even in a computer with less resources



Performance

Figure 28: Web App Performance Report (Google Lighthouse)

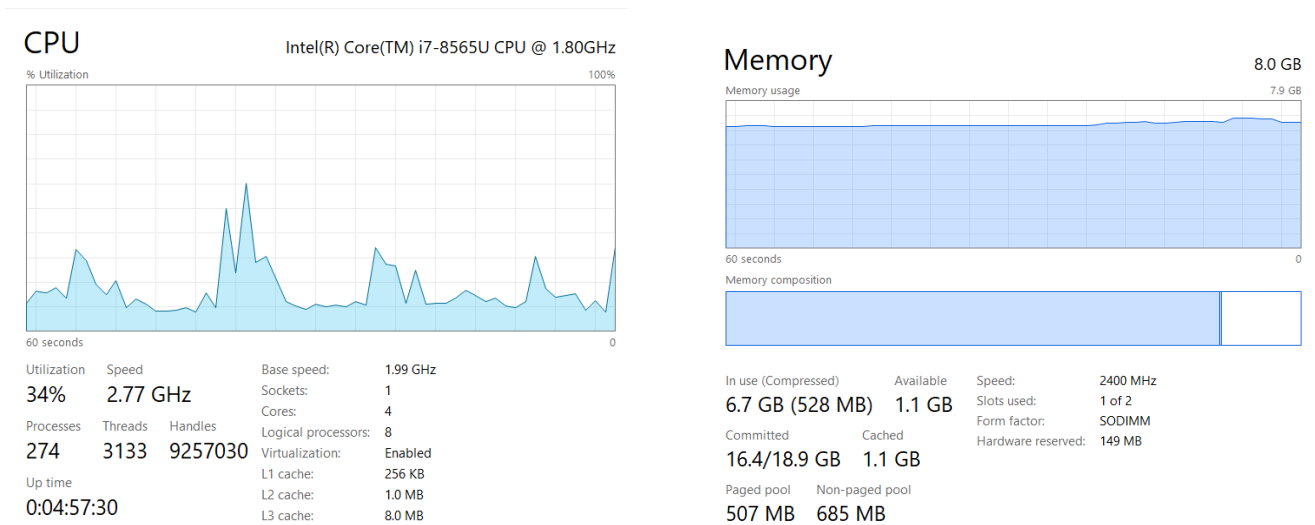


Figure 29: CPU and Memory Usage for Web Application

8.7.2. Usability

Best practices were followed when building the application. A screenshot of the light house report results from googles on the accessibility (93%), best practices (83%) and SEO (90%) levels is attached below.



Figure 30: Web App Usability Report (Google Lighthouse)

This result shows that the author has successfully been able to maintain usability within this research.

8.7.3. Security

The author does not collect any user data within the application. No malware or third-party activities are included within the application. This makes the application highly secure and fulfills the non-functional requirement within this research.

8.7.4. Maintainability

The code of the whole project is available in the following Github repository. github.com/Hamza-Ziyad/SinDOC

The image below shows that the code is in A code quality which makes it highly maintainable. The measurement was done using a code quality tester available online ([CodeFactor](https://codefactor.io)) The author has successfully been able to incorporate maintainability within this research.

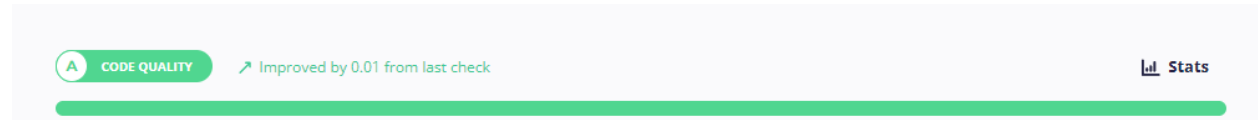


Figure 31: Web App Code Quality Report (CodeFactor)

8.8. Limitations of the Testing Process

There were various limitations faced by the author while working on the testing. There were not many models related to Sinhala summarization to be compared with. Sinhala models do not provide the best results when it comes to ROUGE scores or any other evaluation metrics. Hasan et al. (2021) has also stated this as a previous issue. The limitation on GPU and expenses on them also caused the author to reduce some of the training arguments and not follow some of the SOTA approaches that were provided. The maximum available GPU was used for training this model and testing it. A screenshot of the GPU usage is attached in the **APPENDIX G**. Time limitations in Google Collab also did affect the testing process.

8.9. Chapter Summary

The chapter explains all available testing methods conducted by the author to ensure that they have been applied for both functional and non-functional requirements. Initially the goals and objectives were stated, then the testing criteria's were defined. Then both the model and web application testing results and methods were discussed. The completion status of both the functional and non-functional requirements was discussed. Finally, the limitations of the testing process were explicitly stated.

CHAPTER 09: EVALUATION

9.1. Chapter Overview

This chapter discusses the evaluation steps taken by the author in order to confirm if the author was able to achieve all necessary aims and objectives after implementing the proposed solution. This chapter elaborates on all evaluation approaches, findings, statuses of functional and nonfunctional requirements.

9.2. Evaluation Methodology and Approach

Both qualitative and quantitative approaches were followed in order to evaluate the quality of the project. All quantitative evaluation approaches were stated in the previous chapter while qualitative evaluation results are stated within this chapter. The author conducted interviews in order to qualitatively evaluate the project. The results of the evaluation methods are stated in the upcoming sections.

9.3. Evaluation Criteria

The criteria's followed when evaluating this research project are stated in the table below.

Table 30: Evaluation Criteria

Criteria	Reasoning
Challenges of the domain and research	To understand the amount of importance and impact it has within the domain and to understand how much of a technological contribution it would be to the deep learning industry
Literature review on both domain and technology	To identify how much effort has been put into identifying the research gaps, existing works on both technology and domain and what type of evaluation methodologies were carried out.
Contribution to the problem and research domain	To understand the impact, it has within the Sri Lankan community and to understand how much of a technological contribution it would be for Sinhala summarization tasks.

Model implementation	To ensure that the proposed architecture has been developed and it provides acceptable results.
Appropriate methods on doing quantitative evaluation for Sinhala summarization tasks	To identify and understand if the correct evaluation strategies have been followed while following automatic evaluation strategies.
UI/UX study	Ensure that the general user gets an overall smooth experience within the application
Documentation and code maintenance	Ensure that all research related tests, implementations and documentations are named and stored accordingly.

9.4. Self-Evaluation

The author conducted a self-evaluation in order to ensure that the research satisfies all expected outcomes. The details are stated in the following table.

Table 31: Self-Evaluation Results

Criteria	Self-evaluation
Choice of research	Summarization techniques are widely being implemented for the English language domain due to the increase in storage of data within the internet and less consumption of all data by users. This is a potential problem that would occur for the Sinhala language soon.
Research and problem domain contribution	Technically the research involves hyper parameter tuning that provides improved results for Sinhala summarization tasks. Sri Lankans will be highly benefited with this type of an application as they could get a summary of the article content which are like human written summaries.

Model implementation	Necessary steps for building each extractive, abstractive and hybrid model were highly researched. Use of the right dataset, pre-processing techniques, and pre-trained models were all analyzed and researched to get the best out of them.
Model evaluation	Necessary quantitative evaluation metrics (chapter 8) were calculated to make sure the model performs with high quality in all aspects. And the qualitative evaluations are stated in the following sections.
UI/UX	Following the best UX principles to ensure that the general users are able to use the application in a simple manner. A simple minimalistic design was made to retain users' attention.
Limitations and future work	Domain expert evaluators response was taken into consideration for trying other models and add some new UI elements to enhance user experience within the web application

9.5. Selection of the Evaluators

For conducting the evaluation, the author selected technical experts, domain experts and focus group users. Interviews were done for all the candidates. The number of candidates is stated below. The details of the evaluators are stated in the **APPENDIX H**.

Table 32: Evaluators Count

Expert	Candidate count
Technical	4
Domain	3
General users	6

9.6. Evaluation Result

The themes of the evaluation result are stated in the table below. Proof of evaluators interview responses are attached in the **APPENDIX H**.

Table 33: Evaluation Theme Defining

Theme ID	Theme
TH1	Choice of research
TH2	Research and problem domain contribution
TH3	Model implementation
TH4	Model evaluation
TH5	UI/UX
TH6	Limitations

9.6.1. Technical Experts

Evaluation results and analysis of technical experts are stated in the below table.

Table 34: Technical Expert Evaluation Opinions

Theme ID	Analysis of opinions
TH1, TH2	NLP for the Sinhala language has been growing for the last couple years which makes it a good research area to work on with. Identification of summarization gaps in the Sinhala language domain are well researched.
TH2	The ability to get the right hyper parameters tuned values for summarization makes the process of summarization much easier and faster than manually tuning to get the best parameters from scratch. Publicly hosting the model and datasets in Hugging face for future contributions was also appreciated.
TH3	The implementation of a hybrid system makes the application better because the output will be a result of the best of both models.

TH3	The implementation of a simple light weight prototype with the model has been appreciated in a good way.
TH4	Even though Sinhala language has gaps in evaluation methods. The methods of qualitative and quantitative evaluations conducted for this research have been well attempted with available SOTA approaches for other languages.
TH5	The application is simple to understand for any type of user. The UI design is also very appealing in the aim of getting user attraction. Implementation of alerts was recommended to help users understand the types of summarizations.
TH6	The standard of the model is still not as good as English but does solve the aim appropriately. But improving the model with a better accurate and well-crafted dataset can improve the results massively. Training the model on various another domain dataset was highly recommended too. The application speed of generating a summary needs to be improved.

9.6.2. Domain Experts

Evaluation results and analysis of domain experts are stated in the below table.

Table 35: Domain Expert Evaluation Opinions

Theme ID	Analysis of opinions
TH2	News articles, blogs and other long documents in Sinhala keep increasing within the internet. So, summarizing such long documents can save a lot of hours.
TH3	The results generated from the application are almost grammatically correct. Only in certain cases does the model feed old data to the user. But having an extractive approach was a good way of patching such issues.
TH5	Very simple and user-friendly interface. The accessibility on the web pages is well maintained

TH6	Contents of the website can also be switched to Sinhala as a possible future work. Improving the time taken for summary generation was also stated.
-----	---

9.6.3. Focus Group

Evaluation results and analysis of focus groups or general users are stated in the below table.

Table 36: Focus Group Evaluation Opinions

Theme ID	Analysis of opinions
TH3	Summary generation was meaningful in most cases. Having the opinion of choosing the type of summary to be generated improves the reliability of the application
TH5	Simple to understand and a very user-friendly web application. UI design was appreciated
TH6	Summary results perform well for news data. Making it perform well on other domain documents was suggested as an improvement. The model being up to date with real time data and improving time speed for summary generation was also suggested.

9.7. Limitation of Evaluation

Sinhala language is the main limitation of evaluation. The existing evaluation steps don't work very accurately for Sinhala language as it is a low resource language. All the metrics are highly implemented for English languages to provide the best results. When it comes to Sinhala it's difficult to get the best results during automatic evaluation.

Evaluators specific for summarization tasks were difficult to get in touch with. Even though the author tried to get in touch with foreign evaluators, their responses took a long time. This makes it difficult to evaluate within the project's timeline.

9.8. Evaluation on Functional Requirements

The author was able to achieve 11 out of 14 functional requirements. The functional requirements and the status of implementations are stated in the following table.

Table 37: Functional Requirements Evaluation

FRID	Requirement description	Priority level	Status
FR01	Option of inputting the document text into the system	M	Done
FR02	Option of choosing type of summary that needs to be generated	M	Done
FR03	Verify text	M	Done
FR04	Generate extractive summary	M	Done
FR05	Generate abstractive summary	M	Done
FR06	Generate combined summary with both extractive and abstractive models.	M	Done
FR07	Display the generated summary to user	M	Done
FR08	Alert if no choice of summary is selected	S	Done
FR09	Copy the generate summary to clipboard	S	Done
FR10	Clear the generated summary	S	Done
FR11	Save all generated summaries to a database	C	Not done
FR12	Multiple language summarization support	C	Done
FR13	Input pdf documents	W	Not done
FR14	Summarize images	W	Not done

11/14 (79%) requirements achieved

9.9. Evaluation on Non-Functional Requirements

The author was able to achieve 4 out of 5 non-functional requirements. The non-functional requirements and the status of implementations are stated in the following table.

Table 38: Non-Functional Requirements Evaluation

NFRID	Requirement description	Requirement	Status
NFR01	Implemented system needs to be efficient and effective in generating outputs as quickly as possible.	Performance	Done
NFR02	Implementation should be implemented by considering the user experience they will have within the application. The user interface of the application will support on improving the customer experience	Usability	Done
NFR03	Maintaining users' privacy or data without breaching them should be considered when implementing the proposed system	Security	Done
NFR04	Updating the system accordingly for a longer run could improve the number of consumers. For this the system need to be maintained accordingly and be updated according to latest industry principles	Maintainability	Partially done
NFR05	When the number of consumers is increasing the ability to maintain multiple users without causing errors will be an important aspect. Scaling the system accordingly will reduce these types of errors and provide a better experience	Scalability	Not done
4/5 (80%) requirements achieved			

9.10. Chapter Summary

This chapter elaborates on all evaluation strategies followed by the author in order to check if all research goals have been successfully achieved. Initially the author states the evaluation criteria and then does a thematic analysis on the results provided by the evaluators. The limitations of evaluation steps and methods are also stated. Finally, the completion level of functional and non-functional requirements is mentioned within the chapter.

CHAPTER 10: CONCLUSION

10.1. Chapter Overview

This chapter discusses the final concluding topics of this research project. The knowledge gained from the whole course and modules are stated in this chapter. The new skills gained through this project are also stated. The chapter also speaks on how the author was able to achieve the aims and objectives stated in the first chapter. The limitations and future works for upcoming research are also discussed in this chapter. And then the final topic concludes on how the research has been a valuable contribution to both the research and problem domains.

10.2. Achievements of Research Aim and Objectives

10.2.1. Research Aim

The aim of this research is to design, implement and evaluate a system that does document summarization for a Sinhala document or article and provide a summary of it.

The author was able to achieve the above stated research aim within the given timeframe. The author was able to implement a combined approach of both extractive and abstractive summarization methods to get the best available model for Sinhala summarization. An additional point to note is that even though the author implemented the solution focusing on Sinhala language some of the other high resource language summarization could also be done. The proof of achievement of the aim and objective of this research are stated in the previous two chapters (**Testing an Evaluation**).

10.2.2. Research Objectives

Apart from the aim of the research the author was also able to achieve all research objectives. The status of completion of the research objectives is stated in the following table.

Table 39: Research Objective Completion Status

Research Objective	Description	Status
Problem Identification	This includes identifying a research problem that has enough potential to be solved.	Done

Literature review	This includes in-depth research to assess existing SOTA work	Done
Data gathering and analysis	This includes carrying out data gathering and analyzing them	Done
Research design and implementation	This includes the method of designing the architecture of the solution and implementing it	Done
Testing and evaluation	This includes comparisons with existing models and testing applications.	Done

10.3. Utilization of Knowledge from the Course

The knowledge gained from the degree program or course has immensely contributed to these projects. The modules and their justifications are stated in the following table.

Table 40: Course Modules Utilization

Year	Module	Justification
L4	Programming Principles 1	This module introduced and gave the basic concepts of Python programming language which was immensely used during the core functionality and backend implementations.
L4	Web Design and Development	Basic ideas on Accessibility guidelines and maintaining UX within a web app were taught in this module. Foundation on HTML, CSS and Javascript concepts also helped in building the frontend of the web app.
L4	Computer Science Practice	Foundation on documenting a project's existing works or literature review and the way of presenting solutions to panels were gained.
L5	Software Development	The method of doing a project within the given timeline, and the way of working using Agile methods were all gained through

	Architecture	this module. Correct documentation and following necessary steps of implementing a project are also gained through this module.
L5	Machine Learning and Data Mining	Basic machine learning concepts and ideas were taught in this module.
L6	Applied Artificial Intelligence	Advanced deep learning and machine learning concepts were taught in this module. The aim of building this research was initially identified through this module.
L6	Usability Testing and Evaluation	The method of gathering software requirements specifications and evaluation in the right way by analyzing survey responses, interview responses were gained from this module.

10.4. Use of Existing Skills

The following existing skills were used for successfully completing this research project.

- **UI/UX Design** - The author is a UI/UX enthusiast. This existing skill gained from self-learning and previous internships both helped the author in designing the application by balancing the importance of both the user interface and user experience within the web application.
- **Accessibility foundations** - This skill was mainly gained from internships. The interest was then expanded by learning online courses. That helped the author in maintaining the accessibility within the application by making it accessible to any user within any group of age.
- **NLP** - The author had a particular interest in NLP so online courses and videos were watched before engaging with this research project.
- **Front end development** - The author learned advanced concepts on building responsive and quality web app frontend using Bootstrap and jQuery during internship. This skill was highly utilized when building the HTML templates.

10.5. Use of New Skills

By working on this project, the author was able to gain the following new skills.

- **Summarization** - This project helped the author in acquiring skills on the types of existing summarization methods namely extractive, abstractive and hybrid. The author was able to get a ion depth understanding in each of the summarization approaches.
- **NLP for Sinhala** - Even though the author had a basic understanding on NLP. This project only helped to identify its importance within the Sinhala language domain.
- **Pretrained models** - The importance of building NLP models with existing pretrained models quickly and more efficiently was learned through this research project.

10.6. Achievement of Learning Outcomes

The achievement of learning outcomes of this project module is stated in the following table.

Table 41: Learning Outcome Achievements

Learning Outcome	Description
LO1	All reasonings and justifications on the problem area chosen are attended to within the previous chapters.
LO2	The methodology chapter explains all development and project management steps followed by the author to achieve the research aims
LO3	The architectures of the solution, the design strategies the tools and technologies that were used for this research project are all stated in the previous chapters
LO4	40 plus research, journals and blogs were read and researched. Those were critically evaluated and justified in the above chapters
LO5	Contacting the authors' mentor, gaining reviews from technical, domain experts and focus groups were a crucial part of this project. The author was able to analyze them and successfully complete them
LO6	Datasets, tools, technologies, libraries and models used in this project are fully

	open source. Every aspect of issues was analyzed to align with SLEP rule.
LO7	The various testing methods of multiple models, and system performance while implementing a web application that is smooth for the general audience was highly considered. Related details are stated in the previous chapters.
LO8	The project report was maintained and documented within the timeline.

10.7. Problems and Challenges Faced

The author did come across various problems while undergoing this research project. The problems faced and their mitigations are stated in the following table.

Table 42: Problems and Challenges Faced

Problem	Mitigation
Scarcity of Sinhala data for summarization tasks	The author had to create a sample dataset for training the model. This dataset was a translated version of an English open-source dataset.
Identification of pretrained models that supports Sinhala language	The author had to try various types of models to identify if they would work for Sinhala. This took a lot of time. But eventually the author was able to find a relevant model that supported for Sinhala language.
Summary generation for other domains	The model might generate some inaccurate results for other domain documents as it is trained for news articles. Having the ability of choosing the type of summary they want can solve the problem as they could choose the extractive option for such documents.
Lack of computational resources.	The author had to spend almost \$60 on buying additional computer units for training. This was mainly because training Sinhala language did make use of a lot of GPUS. But finally, after so many purchases the author was somewhat able to build the model.

Limited number of domain experts	There are very few summarizations related domain experts in Sri Lanka. The author had to go for a wider scope of evaluators within the data science field.
Foreign evaluators contact difficulty	Contacting foreign evaluators and getting their response takes a lot of time. The author decided to stick with evaluators within Sri Lanka for this reason.

10.7. Deviations

The research did not have any major deviations. The author initially planned on incorporating an AI solution for the extractive summarization but after research it was identified that the Sinhala language hasn't been trained on much data so it might reduce the quality of the project. So, the author had to stick with a sentence scoring algorithmic approach for the extractive summarization part.

Another thing to note is that even though the author has proposed the solution for any type of Sinhala document summarization, there are less resources for other documents, so the author had to train the model for news articles only.

10.8. Limitations of the Research

The research proposed did have some limitations. Those limitations are stated below.

- The number of trials for hyper parameter tuning to study the best parameters for the model was limited to five as there were computational and time restraints.
- The extractive model had to be a basic algorithmic approach since the dataset used to train isn't fully reliable in generating the best results for extractive summarization.
- Scarcity of real dataset for a Sinhala summarization task made automatic testing and evaluation results not fully accurate.
- ROUGE testing for low resource languages including Sinhala is not accurate during training (Hasan et al., 2021). This makes evaluation ROUGE scores inaccurate.
- No specific models built for Sinhala summarization makes this difficult to do domain benchmarking.

- Sinhala model training requires a lot of GPUS for training. So, this model sometimes had to reduce certain training arguments to reduce the usage of computational resources. The main reason for this is because the author had to spend a lot of pocket money for additional resources.
- The models have been trained for the news article domain. So, the model might generate some inaccurate results for other domain documents.

10.9. Future Enhancements

As this research is openly available for future research it will be a great start point for research on Sinhala summarization techniques.

- The abstractive model can be used to expand on other NLP tasks like text classification, generation (Liu and Lapata, 2019) within the Sinhala language domain.
- Introduction of a question answering system (Zhang, Wei and Zhou, 2019) within this model so that the user could ask questions to explore other content within the document that haven't been generated as a summary.
- Improve the model with a real dataset that contains Sinhala articles and Sinhala summary created by Sinhala domain experts.
- Improve model for other document summarization problem domains for Sinhala language. This model is primarily focused on news articles the author would like to extend to other domains like research articles, legal documents and many more.

10.9. Achievement of the Contribution to the Body of Knowledge

The author was able to successfully contribute to both the summarization technical side and Sinhala language domain by completion of this research project.

10.9.1. Problem Domain

The author was able to introduce a Sinhala summarization model that uses a combined approach of both extractive and abstractive approaches for summarization. This helps the Sri Lankan community generate summaries for long Sinhala documents in the way they want. This makes a great contribution to the problem domain.

10.9.2. Research Domain

The author was able to find the most appropriate way of tuning hyperparameters and use them to train the abstractive model. Additionally, the author was able to build a combined approach of summarization that hasn't been built yet for the Sinhala domain. The author has made a translated sub dataset from the openly available CNN Daily Mail dataset which somewhat addresses the scarcity of data (Hasan et al., 2021). The authors' dataset and abstractive model both are deployed in Huggingface. This makes it open for future researchers to make use of both of and make contributions. This makes a great contribution to the research domain.

10.10. Concluding Remarks

The author was able to build a tool with a combined approach of both extractive and abstractive techniques for a low resource language. The author was able to successfully create an app that summarizes Sinhala documents. The research aim and gaps were addressed in the initial chapters of the research. The existing works, their limitations and future works were then critically evaluated. The project management methodology was then discussed afterwards. The way the author gathered data from both technical experts and general to gather information on identifying the impact of this type of an application with the system was then elaborated. The design chapter then provides all diagrams that relate to building the architecture of the application including the activity flows too. The steps taken for implementing the application and testing the results are then discussed. Afterwards gathering responses from evaluators to check if all necessary requirements have been met and to do self-evaluation to identify if the aim and objectives have been met. And finally, the author speaks about the contribution to the body of knowledge, the limitations of the research and available future works.

REFERENCES

- Akiba, T. et al. (2019). Optuna: A Next-generation Hyperparameter Optimization Framework. Available from <http://arxiv.org/abs/1907.10902> .
- Allahyari, M. et al. (2017). Text Summarization Techniques: A Brief Survey. Available from <http://arxiv.org/abs/1707.02268> .
- Automatic Summarization. (no date). Available from <https://ieeexplore.ieee.org/document/8186895> .
- Carenini, G. and Cheung, J.C.K. (2008). Extractive vs. NLG-based Abstractive Summarization of Evaluative Text: The Effect of Corpus Controversiality. *Proceedings of the Fifth International Natural Language Generation Conference*. June 2008. Salt Fork, Ohio, USA: Association for Computational Linguistics, 33–41. Available from <https://aclanthology.org/W08-1106>.
- Chopra, S., Auli, M. and Rush, A.M. (2016). Abstractive Sentence Summarization with Attentive Recurrent Neural Networks. *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. June 2016. San Diego, California: Association for Computational Linguistics, 93–98. Available from <https://doi.org/10.18653/v1/N16-1012> .
- Deshpande, P. and Jahirabadkar, S. (2021). Study of Low Resource Language Document Extractive Summarization using Lexical chain and Bidirectional Encoder Representations from Transformers (BERT). *2021 International Conference on Computational Performance Evaluation (ComPE)*. December 2021. 457–461. Available from <https://doi.org/10.1109/ComPE53109.2021.9751919>.
- Devlin, J. et al. (2019). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. June 2019. Minneapolis, Minnesota: Association for

- Computational Linguistics, 4171–4186. Available from <https://doi.org/10.18653/v1/N19-1423>.
- DeYoung, J. et al. (2021). MS²: Multi-Document Summarization of Medical Studies. *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. 2021. Online and Punta Cana, Dominican Republic: Association for Computational Linguistics, 7494–7513. Available from <https://doi.org/10.18653/v1/2021.emnlp-main.594>.
- Dhananjaya, V. et al. (2022). BERTifying Sinhala -- A Comprehensive Analysis of Pre-trained Language Models for Sinhala Text Classification. 2022. arXiv. Available from <https://doi.org/10.48550/ARXIV.2208.07864>.
- Digital 2022: Sri Lanka. (2022). *DataReportal – Global Digital Insights*. Available from <https://datareportal.com/reports/digital-2022-sri-lanka>.
- Divya, K. et al. (2020). Text Summarization using Deep Learning. 07 (05).
- Exploring Transfer Learning with T5: the Text-To-Text Transfer Transformer. (2020). Available from <https://ai.googleblog.com/2020/02/exploring-transfer-learning-with-t5.html>.
- Fabbri, A.R. et al. (2019). Multi-News: a Large-Scale Multi-Document Summarization Dataset and Abstractive Hierarchical Model. Available from <http://arxiv.org/abs/1906.01749>.
- Filippova, K. et al. (2015). Sentence Compression by Deletion with LSTMs. *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. 2015. Lisbon, Portugal: Association for Computational Linguistics, 360–368. Available from <https://doi.org/10.18653/v1/D15-1042>.
- Geitgey, A. (2020). Natural Language Processing is Fun! *Medium*. Available from <https://medium.com/@ageitgey/natural-language-processing-is-fun-9a0bff37854e>.
- Goldstein, J. et al. (2000). Multi-Document Summarization By Sentence Extraction. *NAACL-ANLP 2000 Workshop: Automatic Summarization*. 2000. Available from <https://aclanthology.org/W00-0405>.

- Gonçalves, L. (2021). Automatic Text Summarization with Machine Learning — An overview. *luisfredgs*. Available from <https://medium.com/luisfredgs/automatic-text-summarization-with-machine-learning-an-overview-68ded5717a25>.
- Guzmán, F. et al. (2019). The FLORES Evaluation Datasets for Low-Resource Machine Translation: Nepali–English and Sinhala–English. *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. November 2019. Hong Kong, China: Association for Computational Linguistics, 6098–6111. Available from <https://doi.org/10.18653/v1/D19-1632>.
- Hasan, T. et al. (2021). XL-Sum: Large-Scale Multilingual Abstractive Summarization for 44 Languages. *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*. August 2021. Online: Association for Computational Linguistics, 4693–4703. Available from <https://doi.org/10.18653/v1/2021.findings-acl.413>.
- Hsu, W.-T. et al. (2018). A Unified Model for Extractive and Abstractive Summarization using Inconsistency Loss. *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. July 2018. Melbourne, Australia: Association for Computational Linguistics, 132–141. Available from <https://doi.org/10.18653/v1/P18-1013>.
- <https://www.facebook.com/kdnuggets>. (no date). Approaches to Text Summarization: An Overview. *KDnuggets*. Available from <https://www.kdnuggets.com/approaches-to-text-summarization-an-overview.html>.
- Joshi, P. et al. (2020). The State and Fate of Linguistic Diversity and Inclusion in the NLP World. *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. July 2020. Online: Association for Computational Linguistics, 6282–6293. Available from <https://doi.org/10.18653/v1/2020.acl-main.560> [Accessed 4 May 2023].
- Lee, A. (2023). What Are Large Language Models Used For and Why Are They Important? *NVIDIA Blog*. Available from <https://blogs.nvidia.com/blog/2023/01/26/what-are-large-language-models-used-for/>.

- Lee, C.-H. et al. (2022). DOCmT5: Document-Level Pretraining of Multilingual Language Models. *Findings of the Association for Computational Linguistics: NAACL 2022*, 425–437. Available from <https://doi.org/10.18653/v1/2022.findings-naacl.32>.
- Lee, E. et al. (2022). Pre-Trained Multilingual Sequence-to-Sequence Models: A Hope for Low-Resource Language Translation? Available from <https://doi.org/10.48550/arXiv.2203.08850>.
- Lewis, M. et al. (2019). BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension. Available from <http://arxiv.org/abs/1910.13461>.
- Lin, C.-Y. (2004). ROUGE: A Package for Automatic Evaluation of Summaries. *Text Summarization Branches Out*. July 2004. Barcelona, Spain: Association for Computational Linguistics, 74–81. Available from <https://aclanthology.org/W04-1013>.
- Liu, Y. and Lapata, M. (2019). Text Summarization with Pretrained Encoders. *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. November 2019. Hong Kong, China: Association for Computational Linguistics, 3730–3740. Available from <https://doi.org/10.18653/v1/D19-1387>.
- Liu, Y. et al. (2019). RoBERTa: A Robustly Optimized BERT Pretraining Approach. Available from <http://arxiv.org/abs/1907.11692>.
- Liu, Y. et al. (2020). Multilingual Denoising Pre-training for Neural Machine Translation. *Transactions of the Association for Computational Linguistics*, 8, 726–742. Available from https://doi.org/10.1162/tacl_a_00343.
- Mihalcea, R. (2004). Graph-based Ranking Algorithms for Sentence Extraction, Applied to Text Summarization. *Proceedings of the ACL Interactive Poster and Demonstration Sessions*. July 2004. Barcelona, Spain: Association for Computational Linguistics, 170–173. Available from <https://aclanthology.org/P04-3020>.

- Mishra, P. (2021). Understanding T5 Model: Text to Text Transfer Transformer Model. *Medium*. Available from <https://towardsdatascience.com/understanding-t5-model-text-to-text-transfer-transformer-model-69ce4c165023> .
- Mrabet, Y. and Demner-Fushman, D. (2020). HOLMS: Alternative Summary Evaluation with Large Language Models. *Proceedings of the 28th International Conference on Computational Linguistics*. December 2020. Barcelona, Spain (Online): International Committee on Computational Linguistics, 5679–5688. Available from <https://doi.org/10.18653/v1/2020.coling-main.498> .
- Nenkova, A. (2011). Automatic Summarization. *Foundations and Trends® in Information Retrieval*, 5 (2), 103–233. Available from <https://doi.org/10.1561/15000000015>.
- Papineni, K. et al. (2001). BLEU: a method for automatic evaluation of machine translation. *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics - ACL '02*. 2001. Philadelphia, Pennsylvania: Association for Computational Linguistics, 311. Available from <https://doi.org/10.3115/1073083.1073135> .
- [PDF] A Combined Model for Extractive and Abstractive summarization based on Transformer model | Semantic Scholar. (no date). Available from <https://www.semanticscholar.org/paper/A-Combined-Model-for-Extractive-and-Abstractive-on-Liu-Xu/c3e185b4685ecbe09c5860ce021b95e25c0357dc> .
- Rahul, Adhikari, S., and Monika. (2020). NLP based Machine Learning Approaches for Text Summarization. *2020 Fourth International Conference on Computing Methodologies and Communication (ICCMC)*. March 2020. 535–538. Available from <https://doi.org/10.1109/ICCMC48092.2020.ICCMC-00099>.
- Rathnasena, K.A.M.P. et al. (2018). Summarization based approach for Old Sinhala Text Archival Search and Preservation. *2018 18th International Conference on Advances in ICT for Emerging Regions (ICTER)*. September 2018. 182–188. Available from <https://doi.org/10.1109/ICTER.2018.8615585>.

- Sakhadeo, A. and Srivastava, N. (2018). Effective extractive summarization using frequency-filtered entity relationship graphs. Available from <http://arxiv.org/abs/1810.10419> .
- Sandaruwan, H.M.S.T., Lorensuhewa, S.A.S. and Kalyani, M.A.L. (2019). Sinhala Hate Speech Detection in Social Media using Text Mining and Machine learning. *2019 19th International Conference on Advances in ICT for Emerging Regions (ICTer)*. September 2019. 1–8. Available from <https://doi.org/10.1109/ICTer48817.2019.9023655>.
- Steinberger, J. and Ježek, K. (no date). Using Latent Semantic Analysis in Text Summarization and Summary Evaluation.
- Subramanian, S. et al. (2020). On Extractive and Abstractive Neural Document Summarization with Transformer Language Models. Available from <http://arxiv.org/abs/1909.03186> .
- Timalsina, B., Paudel, N. and Shahi, T.B. (2022). Attention based Recurrent Neural Network for Nepali Text Summarization. *Journal of Institute of Science and Technology*, 27 (1), 141–148. Available from <https://doi.org/10.3126/jist.v27i1.46709>.
- Tretyak, V. and Stepanov, D. (2020). Combination of abstractive and extractive approaches for summarization of long scientific texts. Available from <http://arxiv.org/abs/2006.05354>
- Veenadhari, G.V. and Bharathi, B.R. (2022). IMPROVING HYBRID SUMMARIZATION BY USING ABSTRACT AND EXTRACT MODEL. 10 (9).
- Wan, D. and Bansal, M. (2022). FactPEGASUS: Factuality-Aware Pre-training and Fine-tuning for Abstractive Summarization. Available from <https://doi.org/10.48550/arXiv.2205.07830>
- What is a managed database? (no date). Available from <https://www.oracle.com/hk/artificial-intelligence/what-is-natural-language-processing/>
- What is Natural Language Processing? | IBM. (no date). Available from <https://www.ibm.com/topics/natural-language-processing>
- Wolf, T. et al. (2020). Transformers: State-of-the-Art Natural Language Processing. *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System*

- Demonstrations*. October 2020. Online: Association for Computational Linguistics, 38–45. Available from <https://doi.org/10.18653/v1/2020.emnlp-demos.6>
- Xue, L. et al. (2021). mT5: A massively multilingual pre-trained text-to-text transformer. Available from <http://arxiv.org/abs/2010.11934>
- Yu, B. (2022). Evaluating Pre-Trained Language Models on Multi-Document Summarization for Literature Reviews. 2022. Available from <https://www.semanticscholar.org/paper/Evaluating-Pre-Trained-Language-Models-on-for-Yu/899539650b823935247d74c7fdd62e98866df2b0>.
- Zhang, J. et al. (2020). PEGASUS: Pre-training with Extracted Gap-sentences for Abstractive Summarization. Available from <http://arxiv.org/abs/1912.08777>
- Zhang, X., Wei, F. and Zhou, M. (2019). HIBERT: Document Level Pre-training of Hierarchical Bidirectional Transformers for Document Summarization. Available from <http://arxiv.org/abs/1905.06566>
- Zhou, L. and Hovy, E. (2004). Template-Filtered Headline Summarization. *Text Summarization Branches Out*. July 2004. Barcelona, Spain: Association for Computational Linguistics, 56–60. Available from <https://aclanthology.org/W04-1010>
- Zhu, J. et al. (2019). NCLS: Neural Cross-Lingual Summarization. *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. November 2019. Hong Kong, China: Association for Computational Linguistics, 3054–3064. Available from <https://doi.org/10.18653/v1/D19-1302>
- Doshi, K. (2021). Foundations of NLP Explained — Bleu Score and WER Metrics. *Medium*. Available from <https://towardsdatascience.com/foundations-of-nlp-explained-bleu-score-and-wer-metrics-1a5ba06d812b> .

APPENDIX A: IN-SCOPE, OUT-SCOPE AND PROTOTYPE DIAGRAM

In-scope

All sections that will be covered in the research are stated as follows.

- Reviewing models that have been trained for document summarization tasks.
- Reviewing the use of summarizing documents and articles in Sinhala.
- Evaluating the project with the help of domain experts and gathering their feedback to create a better solution.
- Creating a web app that takes the document content as an input and provides a summarized version of the document in Sinhala.

Out-scope

All sections that will not be covered in the research are stated as follows.

- Text scans from images will not be included in the project.
- Voice-to-text feature will not be included in the project.

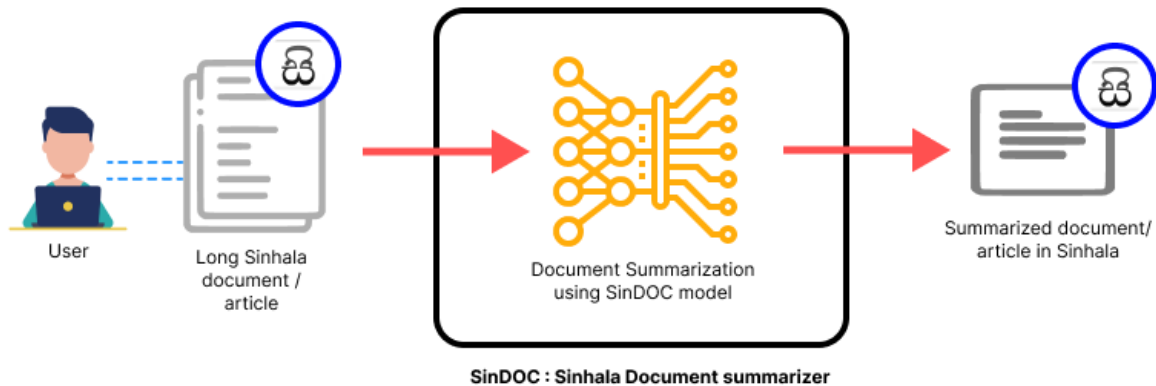


Figure 32: SinDOC Feature Prototype Diagram (Self-composed)

Diagram depicting the prototype feature.

APPENDIX B: CONCEPT MAP

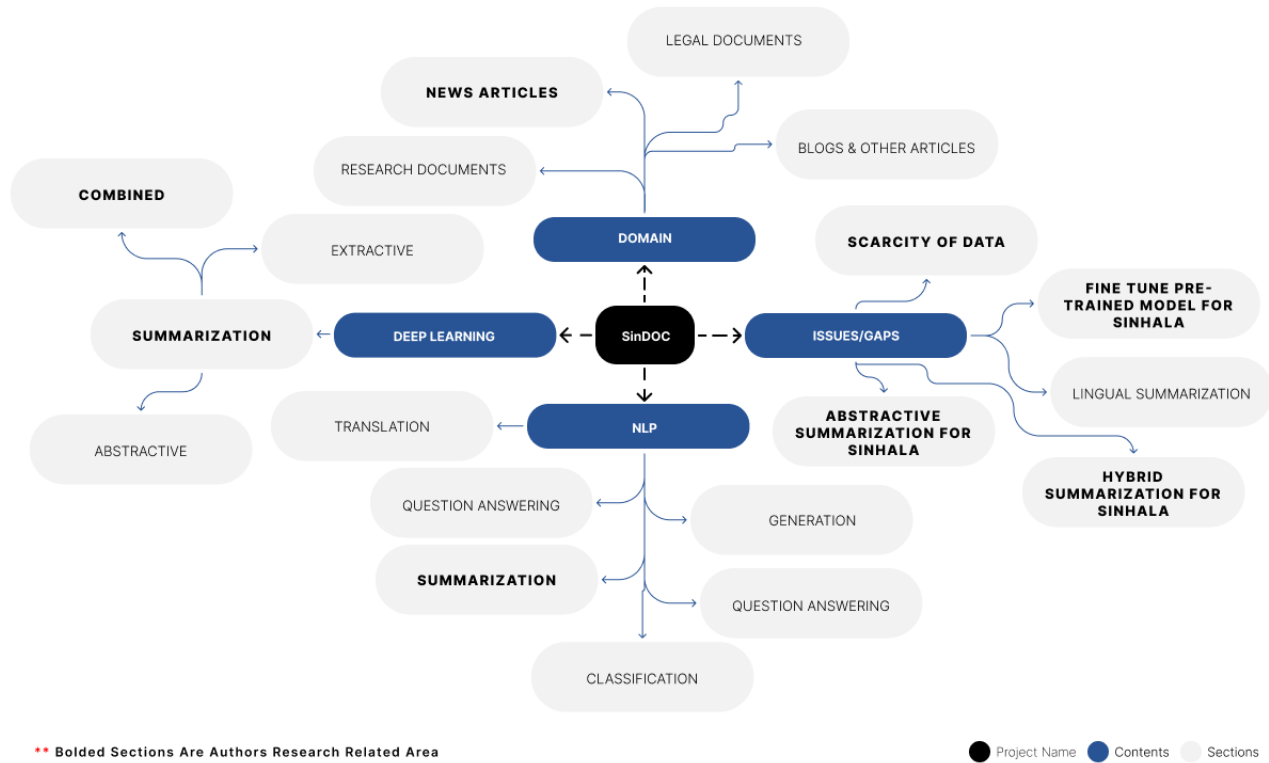


Figure 33: Concept map (Self-composed)

APPENDIX C: GANTT CHART

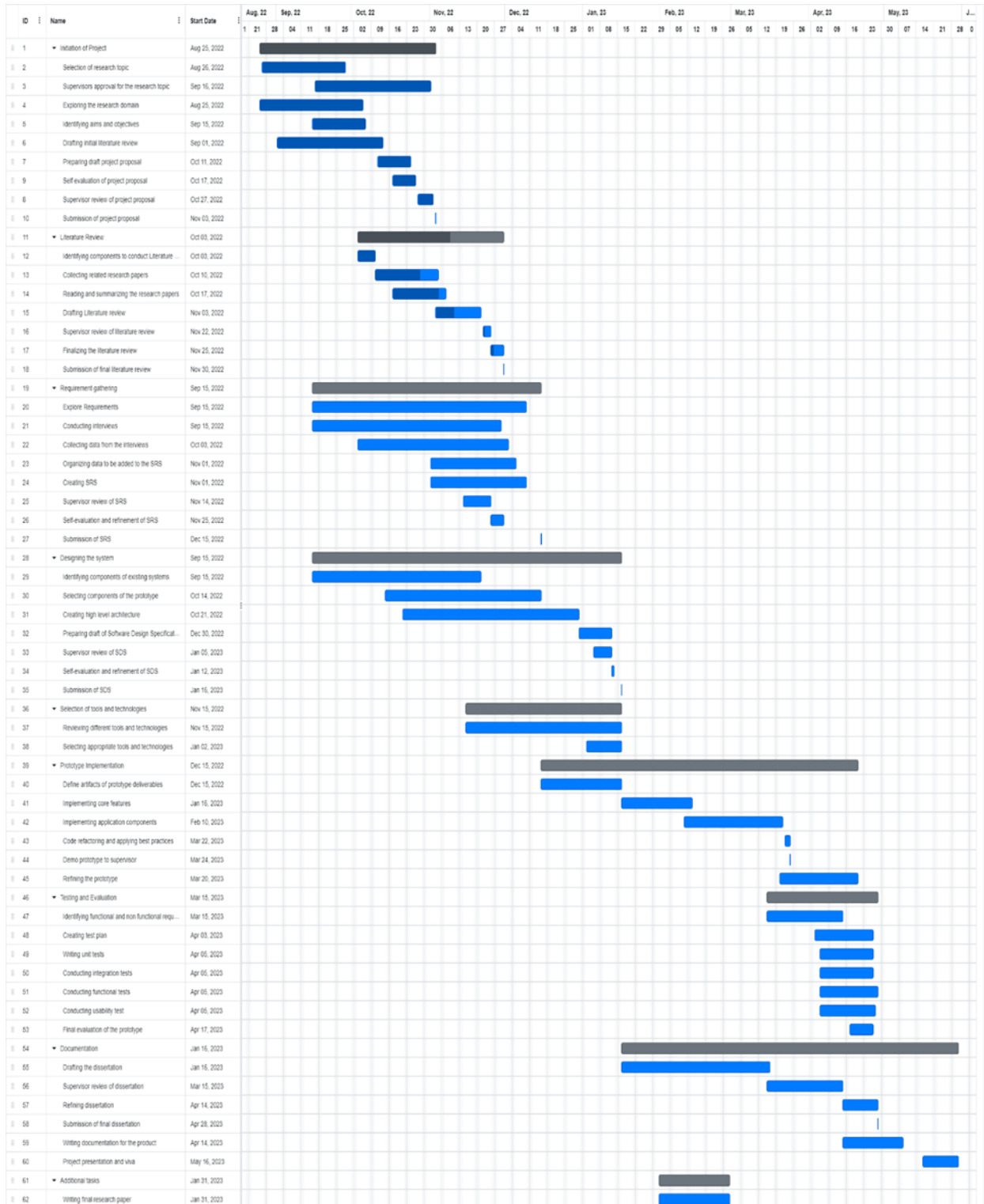


Figure 34: Gantt Chart (Self-composed)

APPENDIX D: SRS

Questions

Responses 112

Settings

SinDOC : A summarization tool for Sinhala documents

I'm Hamza Ziyad, a BSc. (Hons) Computer Science undergraduate in my final year at the Informatics Institute of Technology. I would appreciate if you could take a few minutes to complete the questionnaire below so that I can get the necessary information that I need to further carry out my study.

This survey is conducted to gather data for my final year project "SinDOC". The ultimate goal of the research is to generate accurate and simple summaries for long Sinhala documents.

Note: All information collected will be used for academic purposes only, and all information provided will be treated as confidential and will remain anonymous.

hamza.2019407@iit.ac.lk [Switch account](#)

Not shared

* Indicates required question

System Overview

The user will have to input a long Sinhala document and the system will generate a simple summary of the document.

Preview

```
graph LR; User((User)) --> Input[Long Sinhala document / article]; Input --> Model[Document Summarization using SinDOC model]; Model --> Output[Summarized document article in Sinhala];
```

SinDOC: Sinhala Document Summarizer

Are you a person who reads Sinhala documents ? *

☐ Yes

☐ No

☐ Maybe

What type of Sinhala documents do you read? *

☐ News articles

☐ Blogs & researches

☐ Legal documents

☐ Other documents

Figure 35: SRS Survey -Page 1

How often do you read Sinhala documents? *

1 2 3 4 5

Very rarely ☐ ☐ ☐ ☐ ☐ Very often

How much do you complete when reading such documents? *

☐ 0%

☐ 25%

☐ 50%

☐ 75%

☐ 100%

Do you get bored when reading long Sinhala documents? *

☐ Yes

☐ Very often

☐ Rarely

☐ Not at all

Would you like to get a summarized version of your document in Sinhala? *

☐ Yes

☐ Maybe

☐ No

Have you used any Sinhala summarization tools before? *

☐ Yes

☐ No

If yes, how accurate do you think they are ?

1 2 3 4 5

Very less accurate ☐ ☐ ☐ ☐ ☐ Very highly accurate

Figure 36: SRS Survey -Page 3

How useful do you think a Sinhala document summarization tool would be for you *

1 2 3 4 5

Very useless ☐ ☐ ☐ ☐ ☐ Very useful

Would you prefer to have a model that summarizes many languages including sinhala? *

☐ Yes

☐ No i would want only for Sinhala

☐ I have no specific choice , anything would be fine

Submit Clear form

This form was created inside of Informatics Institute of Technology. [Report Abuse](#)

Google Forms

Figure 37: SRS Survey -Page 2

Table 43: SRS Formal Interview Thematic

Theme	Evidence
Dataset collection	<p><i>“Conduct a survey for a group of people who reads Sinhala documents.”</i></p> <p><i>“Use their responses to identify the best dataset”</i></p>
Data preparation and pre-processing	<p><i>“Most tokenizers are UTF-8 which won’t work for Sinhala”</i></p> <p><i>“Try sinling a SinalaTokenizer library”</i></p>
Limitations of summarization techniques for low resource languages	<p><i>“Provide users options of choosing type of summary to be generated would be better.”</i></p> <p><i>“Try using available datasets and train, refrain from creating one unless you really have to.”</i></p> <p><i>“Try sinling a SinalaTokenizer library”</i></p>
Model development	<p><i>“Use of pre-trained models such as T5, Bard, Bert can help you save resources and provide a quality output.”</i></p> <p><i>“Generate summaries with the combination of both extractive and abstractive techniques.”</i></p> <p><i>“Make sure the UI is minimal and user friendly”</i></p>
Evaluation techniques	<p><i>“Don't stick with ROGUE only, try other testing methods too.”</i></p> <p><i>“Do a evaluation with language experts after building the model”</i></p>

Table 44: UC1 Use Case Description

Use Case Name	Enter Sinhala document content
Use Case ID	UC1
Description	The user will have to input the document content that needs to be summarized.
Priority	High
Actors	User
Pre-conditions	The document needs to have a minimum of two sentences or more.
Extended use cases	Verify text
Included use cases	Choose type of summary
Main flow	<ol style="list-style-type: none"> 1. User inputs Sinhala document 2. System checks if the document is valid. 3. If document is valid it will send into model or an error message will be prompted 4. Generates summary of the Sinhala document
Alternative flow	None
Exceptional flows	Displays error message if document text is invalid
Post conditions	The user should be able to get a summary in Sinhala of the valid document

Table 45: UC3 Use Case Description

Use Case Name	Generate summary
Use Case ID	UC3

Description	The model will be generated a summary according to users document and choice of summary type
Priority	High
Actors	User
Pre-conditions	The document needs to be with a lot of sentences for a better result
Extended use cases	None
Included use cases	Display generated summary
Main flow	<ol style="list-style-type: none"> 1. Identifies user choice of summary. 2. System checks if the type of summary is checked. 3. System generates summary. 4. Generates a summary of the Sinhala document inputted according to the selected type of summary.
Alternative flow	In step 3, if the user chooses extractive model system generates extractive summary, if users choose abstractive model system generates abstractive summary or if user chooses combined model system generates combined summary with use of extractive and abstractive methods
Exceptional flows	In step 3, if the extractive option is chosen and the number sentences are less with high sentence scores then the output might return the same result as the input.
Post conditions	The user should be able to get a summary of the document according to their choice of summary

APPENDIX E: DESIGN

Low Fidelity Prototypes

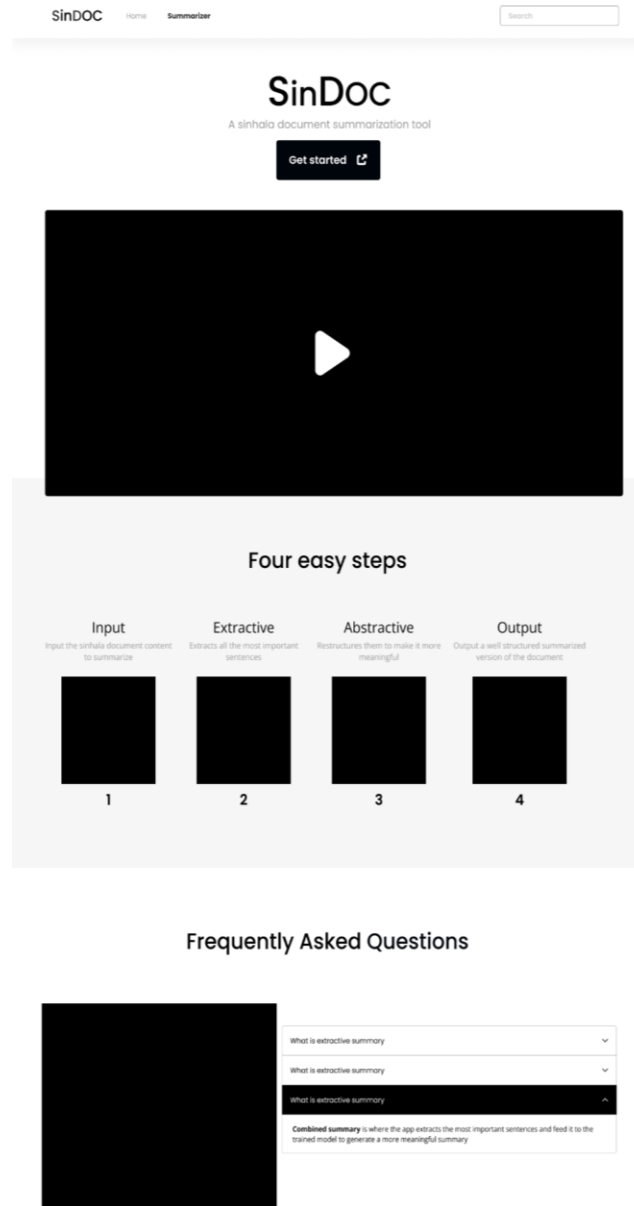


Figure 38: SinDOC Home (Wireframe)

SinDOC

Home

Summarizer

Search

Enter document to summarize :

Enter your document here

Generated summary :

☐ Extractive ☐ Abstractive ☐ Combined

Generate summary

Figure 39; SinDOC Summarizer (Wireframe)

High Fidelity Prototypes

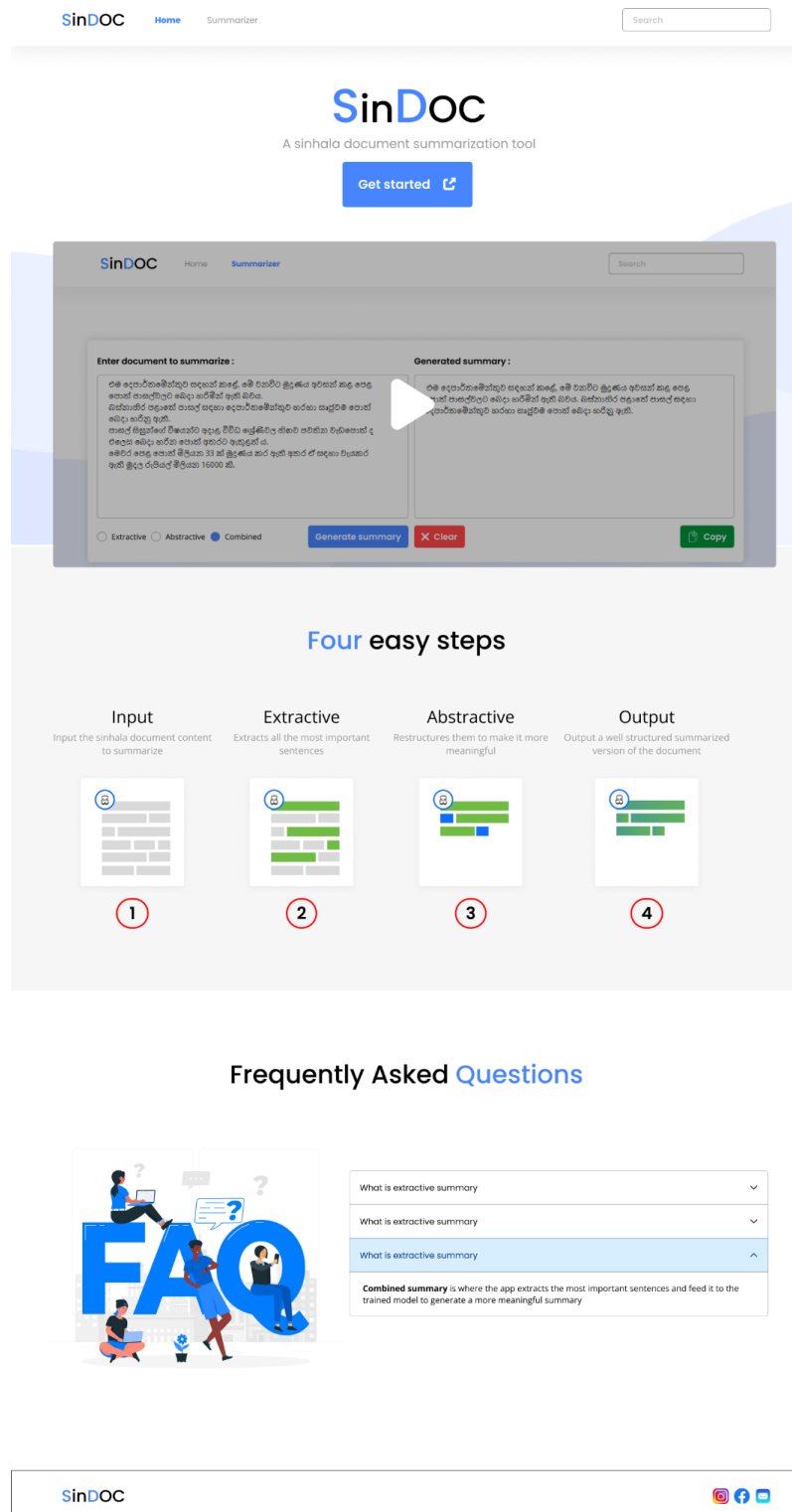


Figure 40: SinDOC Home (Prototype)

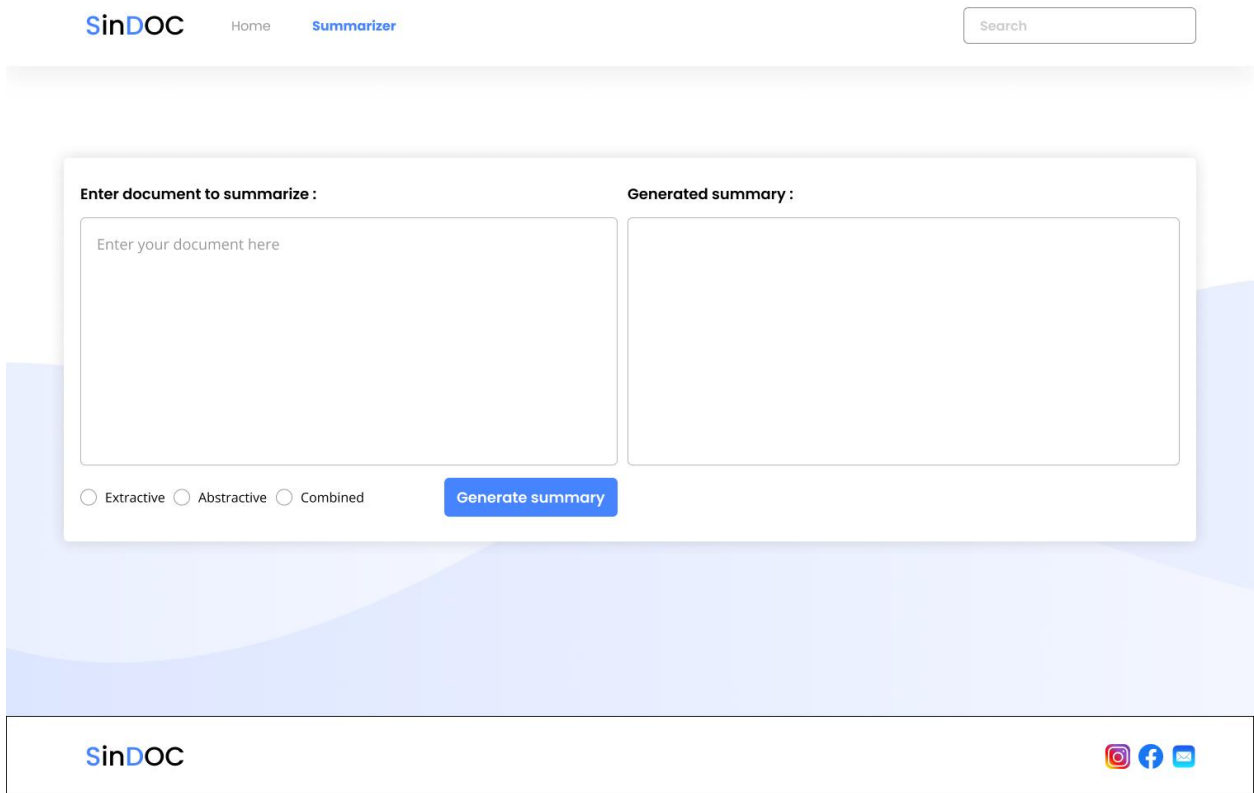


Figure 41 SinDOC Summarizer Before (Prototype)

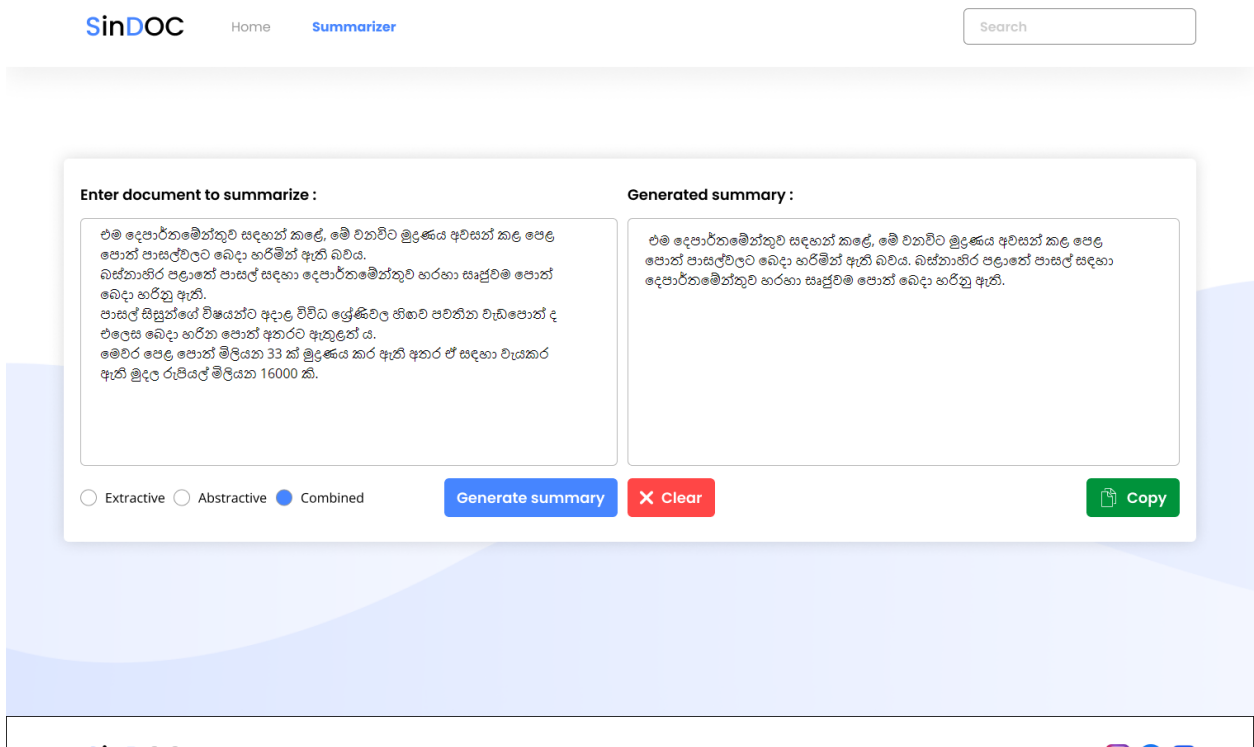





Figure 42: SinDOC Summarizer After (Prototype)


APPENDIX F: IMPLEMENTATION


Dataset Creation


 **Hugging Face**


 Models

 Datasets

 **Datasets:**


 Hamza-Ziyad / **CNN-Daily-Mail-Sinhala**




 like


0

Tasks:


 Summarization

Languages:


 Sinhala


 English


Size Categories:


 1K<n<10K


Tags:


 sinhala-summarization


 abstractive

 extractive

 Dataset card

 Files and versions

 Community

 Settings

Dataset Preview

Size: 106 MB

 API

 Go to dataset viewer

Split

train (6k rows)

article (string)	highlights (string)	id (string)	summary_sinhala (string)	article_sinhala (string)
"LONDON, England (Reuters) -- Harry...	"Harry Potter star Daniel Radcliffe...	"42c027e4ff9730fbb3de84c1af0d2c506e41c3e4"	"හැරි පොටර් නරුඹ දැක්වූ රූපයක්..."	"ලන්ඩන්, එංගලන්තය (රොයිටර්) -- හැරි..."
"Editor's note: In our Behind the Scenes...	"Mentally ill inmates in Miami..."	"ee8871b15c50d0db17b0179a6d2beab35065f1e9"	"මියැම් හි මානසික ආබාධිත රැඳවියන්..."	"කර්තෘගේ සටහන: අපගේ Behind the..."
"MINNEAPOLIS, Minnesota (CNN) --...	"NEW: 'I thought I was going to die,..."	"06352019a19ae31e527f37f7571c6dd7f0c5da37"	"අලුත්: 'මම හිතුවා මම මැරෙයි කියලා,..."	"මිනිසාපොලීස්, මිනිසෙව්වා (සිළුමනින්) ..."
"WASHINGTON (CNN) -- Doctors removed five...	"Five small polyps found during..."	"24521a2abb2e1f5e34e6824e0f9e56904a2b0e88"	"කිරියා පරිපාටිය අතරතුර කුඩා පොලිප්..."	"මොසිංටන් (සිළුමනින්) - වෛද්‍යවරුන් විසින්..."
"BAGHDAD, Iraq (CNN) - Dressed in a...	"Parents beam with pride, can't stop..."	"a1ebbb8bb4d370a1fdf28769206d572be60642d70"	"දෙමව්පියන් ආඩම්බරයෙන්..."	"බැග්දාද්, ඉරාකය (සිළුමනින්) -- සුපර්මැ..."
"BOGOTA, Colombia (CNN) -- A key rebel...	"Tomas Medina Caracas was a..."	"f0d73bdab711763e745cdc75850861c9018f235d"	"වොමස් මෙඩිනා කාරකයේ යනු එක්සත්..."	"බොගෝටා, කොලොම්බියා..."
"WASHINGTON (CNN) -- White House press...	"President Bush says Tony Snow..."	"5e22bbfc7232418bd2dd646b952e404df5bd048"	"ජනාධිපති බුෂ් පවසන්නේ ටෝනි ස්..."	"මොසිංටන් (සිළුමනින්) - පිළිකාවක් සඳහා..."

< Previous

1

2

3

...

60

Next >

Dataset Summary

This dataset card aims to be creating a new dataset or Sinhala news summarization tasks. It has been generated using [\[https://huggingface.co/datasets/cnn_dailymail\]](https://huggingface.co/datasets/cnn_dailymail) and google translate.

Figure 43: Deployed Sinhala Translated Dataset

Data Preprocessing

```
# Drop the title, url, and id columns
df_test = df_test.drop(["article", "id", "highlights"], axis=1)
df_test.head()
```

Figure 46: Removal of Unwanted Columns

```
# Remove the word "CNN" from the summary_sinhala column
df_test["summary_sinhala"] = df_test["summary_sinhala"].str.replace("CNN", "")
# Remove the word "(සිළුමිණි)" from the summary_sinhala column
df_test["summary_sinhala"] = df_test["summary_sinhala"].str.replace("(සිළුමිණි)", "")

# Remove the word "CNN" from the article_sinhala column
df_test["article_sinhala"] = df_test["article_sinhala"].str.replace("CNN", "")
# Remove the word "(සිළුමිණි)" from the article_sinhala column
df_test["article_sinhala"] = df_test["article_sinhala"].str.replace("(සිළුමිණි)", "")

# Updated dataset
df_test.head()
```

Figure 45: Removal of the word CNN

```
# Drop the rows with missing values from text col
dataset_text_col = dataset_text_col.dropna()
# Drop the rows with missing values from summary col
dataset_summary_col = dataset_summary_col.dropna()
```

Figure 44: Remove Rows with Null Values

```
import re

# Define the regular expression pattern for Markdown links
link_pattern = re.compile(r'\[[^\]]+\]\((https?://[^\)]+)\)')

# Define a function to remove Markdown links from a text string
def remove_links(text):
    return link_pattern.sub(r'\1', text)

# Apply the function to the "text" and "summary" columns in the DataFrame
df_test['article_sinhala'] = df_test['article_sinhala'].apply(remove_links)
df_test['summary_sinhala'] = df_test['summary_sinhala'].apply(remove_links)
```

Figure 47: Removal of URL links

```

# Define the regular expression pattern for HTML tags
tag_pattern = re.compile(r'<[^>]+>')

# Define a function to remove HTML tags from a text string
def remove_tags(text):
    return tag_pattern.sub('', text)

# Apply the function to the "text" and "summary" columns in the DataFrame
df_test['article_sinhala'] = df_test['article_sinhala'].apply(remove_tags)
df_test['summary_sinhala'] = df_test['summary_sinhala'].apply(remove_tags)

```

Figure 48: Removal of HTML Tags

```

# Define a function to reduce consecutive words
def reduce_consecutive_words(text):
    words = text.split()
    reduced_words = [words[0]]
    for word in words[1:]:
        if word != reduced_words[-1]:
            reduced_words.append(word)
    return ' '.join(reduced_words)

# Apply the function to the "text" and "summary" columns in the DataFrame
df_test['article_sinhala'] = df_test['article_sinhala'].apply(reduce_consecutive_words)
df_test['summary_sinhala'] = df_test['summary_sinhala'].apply(reduce_consecutive_words)

```

Figure 49: Removal of Consecutive Duplicate words

Abstractive Model

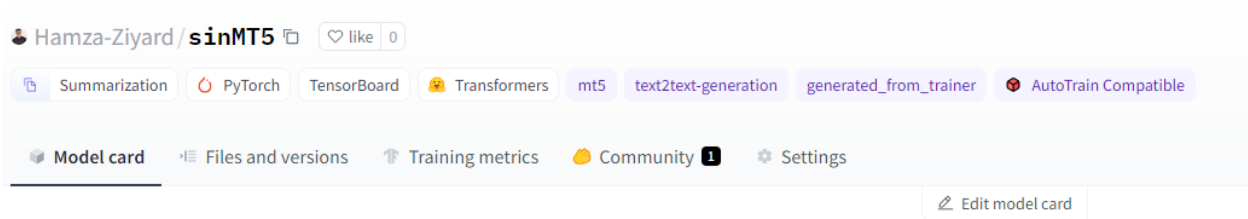


Figure 50: sinMT5 deployed model

```
[I 2023-05-07 00:00:00,946] A new study created in memory with name: no-name-30799a64-a3b6-4f4c-98a6-e87a74ec4d0
[2023-05-07 00:00:00,946] Trial 0 finished with value: 2.037352991104126 and parameters: {'weight_decay': 0.011690798218645e-05, 'learning_rate': 0.0001565224986158822, 'num_train_epochs': 7, 'warmup_ratio': 0.01485252881736892, 'per_device_eval_batch_size': 4, 'per_device_train_batch_size': 16}
[2023-05-07 00:00:00,946] Trial 1 finished with value: 2.037352991104126 and parameters: {'weight_decay': 0.00793629280142335, 'learning_rate': 0.00121675863484e-05, 'num_train_epochs': 6, 'warmup_ratio': 0.02435689973885924, 'per_device_eval_batch_size': 4, 'per_device_train_batch_size': 8}
[2023-05-07 00:00:00,946] Trial 2 finished with value: 2.040651798564879 and parameters: {'weight_decay': 0.0009675800621889354, 'learning_rate': 0.00018455517544832381, 'num_train_epochs': 8, 'warmup_ratio': 0.05449318622889982, 'per_device_eval_batch_size': 8, 'per_device_train_batch_size': 16}
[2023-05-07 00:00:00,946] Trial 3 finished with value: 2.037352991104126 and parameters: {'weight_decay': 5.6328826559139674e-05, 'learning_rate': 0.001845931884264495, 'num_train_epochs': 6, 'warmup_ratio': 0.022872325937985, 'per_device_eval_batch_size': 4, 'per_device_train_batch_size': 16}
[2023-05-07 00:00:00,946] Trial 4 finished with value: 2.037352991104126 and parameters: {'weight_decay': 0.000122083889778879, 'learning_rate': 0.00018478287559372, 'num_train_epochs': 5, 'warmup_ratio': 0.001400018779477945, 'per_device_eval_batch_size': 4, 'per_device_train_batch_size': 16}

print(params.best_params)
{'weight_decay': 0.011690798218645e-05, 'learning_rate': 0.0001565224986158822, 'num_train_epochs': 7, 'warmup_ratio': 0.01485252881736892, 'per_device_eval_batch_size': 4, 'per_device_train_batch_size': 4}
```

Figure 51: Automatically Generated Training Arguments for each Trial

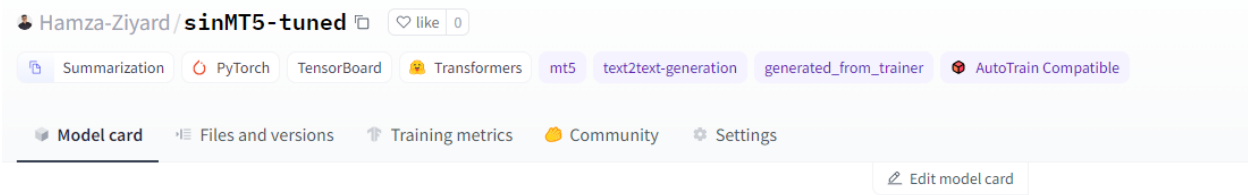


Figure 52: sinMT5-tuned deployed model

```

#Now we have our model so we can summarise our summary
from transformers import AutoTokenizer, AutoModelForSeq2SeqLM

tokenizer = AutoTokenizer.from_pretrained("Hamza-Ziyad/sinMT5-tuned")

model = AutoModelForSeq2SeqLM.from_pretrained("Hamza-Ziyad/sinMT5-tuned")

def abstractive_summarizer(text, model=model):
    text_encoding = tokenizer(
        text,
        max_length=900,
        padding='max_length',
        truncation=True,
        return_attention_mask=True,
        add_special_tokens=True,
        return_tensors='pt'
    )
    #taking out generated ids according to model saved
    generated_ids = model.generate(
        input_ids=text_encoding['input_ids'],
        attention_mask=text_encoding['attention_mask'],
        max_length=150,
        num_beams=4,
        repetition_penalty=2.5,
        length_penalty=1.0,
        early_stopping=True
    )
    #taking out the predictions
    preds = [
        tokenizer.decode(gen_id, skip_special_tokens=True, clean_up_tokenization_spaces=True)
        for gen_id in generated_ids
    ]
    #returning the predictions
    return "".join(preds)

```

Figure 53: Abstractive summarizer function (Abstractive)

APPENDIX G: TESTING

Extractive scores

[illegible]

Figure 55: ROUGE scores on BBC test data (Extractive)

[illegible]

Figure 54: ROUGE scores on CNN test data (Extractive)

Abstractive scores

```

('route-1', 'f', 0.8268767629708129, 'f', 0.69585253462212, 'f', 0.8268767629708129, 'route-2', 'f', 0.458163359116022, 'f', 0.384255925952924, 'f', 0.48131603111335, 'route-1', 'f', 0.8, 'f', 0.4935487470967744, 'f', 0.4561403807192961)
('route-1', 'f', 0.8359764641123, 'f', 0.6444692241048, 'f', 0.8359764641123, 'route-2', 'f', 0.4322224441048, 'f', 0.33521350508134, 'f', 0.37761305080134, 'route-1', 'f', 0.5073999140254, 'f', 0.31495460173025, 'f', 0.441293122404954)
('route-1', 'f', 0.815214179053494, 'f', 0.674808149053494, 'f', 0.815214179053494, 'route-2', 'f', 0.41961490730724, 'f', 0.350351370297226, 'f', 0.38103455666076, 'route-1', 'f', 0.54423677840408, 'f', 0.4060261376227254, 'f', 0.40606082566234)
('route-1', 'f', 0.793778164840276, 'f', 0.721598191797939, 'f', 0.793778164840276, 'route-2', 'f', 0.40651731208047, 'f', 0.368275627545087, 'f', 0.386275627545087, 'route-1', 'f', 0.450241555330807, 'f', 0.351268147724381)
('route-1', 'f', 0.82131614096356, 'f', 0.69087259567442, 'f', 0.82131614096356, 'route-2', 'f', 0.413747807607314, 'f', 0.36497024259541, 'f', 0.3830416432937934, 'route-1', 'f', 0.54423677840408, 'f', 0.4060261376227254, 'f', 0.40606082566234)
('route-1', 'f', 0.822513341096356, 'f', 0.69087259567442, 'f', 0.822513341096356, 'route-2', 'f', 0.413747807607314, 'f', 0.36497024259541, 'f', 0.3830416432937934, 'route-1', 'f', 0.54423677840408, 'f', 0.4060261376227254, 'f', 0.40606082566234)
('route-1', 'f', 0.79459427537861, 'f', 0.7120049438112354, 'f', 0.79459427537861, 'route-2', 'f', 0.405261085923409, 'f', 0.3589070809514, 'f', 0.384529173745443, 'f', 0.33936660304343, 'f', 0.508017453831241, 'f', 0.519377047800122)
('route-1', 'f', 0.795877862562495, 'f', 0.7120049438112354, 'f', 0.795877862562495, 'route-2', 'f', 0.4157947752129, 'f', 0.37324608769545, 'f', 0.3964378165082596, 'f', 0.35782144141301, 'f', 0.58525604614770, 'f', 0.5121685054916123)
('route-1', 'f', 0.8359764641123, 'f', 0.6444692241048, 'f', 0.8359764641123, 'route-2', 'f', 0.4322224441048, 'f', 0.33521350508134, 'f', 0.37761305080134, 'route-1', 'f', 0.5073999140254, 'f', 0.31495460173025, 'f', 0.441293122404954)
('route-1', 'f', 0.803595973007746, 'f', 0.70026720810436, 'f', 0.803595973007746, 'route-2', 'f', 0.427077652076712, 'f', 0.381467469750750, 'f', 0.3978315214514793, 'route-1', 'f', 0.54423677840408, 'f', 0.4060261376227254, 'f', 0.40606082566234)

```

Figure 56: ROUGE scores on BBC test data (Abstractive)

[illegible]

Figure 57: ROUGE scores on CNN test data (Abstractive)

Combined scores

Combined scores

[illegible]

Figure 59: ROUGE scores on BBC test data (Combined)

[illegible]

Figure 58: ROUGE scores on CNN test data (Combined)

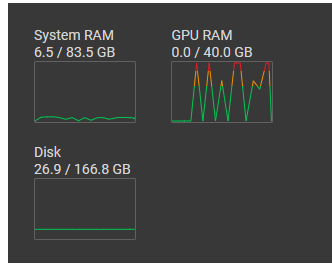


Figure 60: GPU Usage while Training Model (Abstractive)

APPENDIX H: EVALUATION

Table 46: Evaluators Responses

Evaluators	Responses
Technical Experts	
Anonymous Data Scientist Surge Global Pvt. Ltd	<i>“The research gap identified and the contribution that could be made to the Sinhala domain is massive. I have recently come across many Sinhala NLP models that are trying to improve as good as English models. This makes the project a good contribution to the Sinhala domain experts as it is an active research area. I cannot exactly say if the testing process is fully accurate but through the explanation given during evaluation the attempt made in testing the models seems correct. ”</i>
Anonymous Zorro Sign Inc.	<i>“Use of pre-trained models is highly appreciated as no one currently builds anything from scratch. The amount of data fed into such models is also massive so the model automatically tends to provide better results. Making the model publicly available in Huggingface was a good move as most of the NLP researchers use those models for their contributions. It is better to mention that there are not many models to compare with and the testing has been done totally according to your assumptions. And even though the model has been trained through a translated dataset it is always better to train using a dataset that is specifically made for Sinhala. Try that in the future. Keep up the good work.”</i>
Mr. Abdul Haady Senior Software Engineer Calcey Technologies	<i>“Summarization is one of the tasks that every IT industry tries to attend to. Clients nowadays want to convey their message in short words rather than explaining everything in a lengthy document or article. This type of application can help them do so. The problem</i>

Pvt. Ltd	<i>solves not only the Sinhala language domain but also in some of the other languages. Use of a multilingual pre-trained model to do this task is highly appreciated. The model has been well tested as of my knowledge. I would suggest gathering human evaluations rather than solely depending on automatic evaluation methods. “</i>
Umar Mahmoodh UI/UX Designer Dev4s Pvt Ltd.	<i>“During my first view of the web app itself I was able to identify what the application was about. This states that you have used the UX principles effectively within the website. I was able to understand the whole navigation easily. The contents and sections are well separated and designed. I would only suggest having a collapsible banner or an info popup in the app page so that the users will be able to understand what extractive, abstractive or combined really means. Overall, the accessibility, UI and UX are well thought out.”</i>
Domain Experts	
Ms, Damayanthi O/L Sinhala Teacher Zahira College	<i>“Very well-done application. It could save a lot of time for people who read long Sinhala newspapers and articles. There are certain contents that are not grammatically correct, but the overall summary generated can be easily understood by the Sinhalese. By time I think the application could be expanded in a much more efficient way.”</i>
Ms, Nadeeshani A/L Sinhala Teacher Zahira College	<i>“Nice work on a Sinhala summarization application. Very clean design and easily understandable navigation. The only additional option I would like to add to the app is to have an option of translating its content to Sinhala. Sometimes there are certain spaces generated in between letters of words. But most of the Sinhala readers will be able to ignore that as they can still understand the content. It might only be difficult for new learners. I</i>

	<i>hope this issue can be fixed in the future. Having the option of choosing the type of summary reduces those errors most of the time, but you can't rely on that mostly because every user won't understand what they mean.”</i>
Mohamed Razeen Anonymous workplace	<i>“As my workplace is related to the news industry, we always think on how to pass the message to the community as quickly and as short as possible. One of the most difficult parts of our job is to shorten long news reports. An app like this will help us automate that process. I think the app is not 100% capable of going into an industry level for now but it is a very good introduction for future enhancements. Do not lose hope keep improving”</i>
Focus group	
Mr. Suhith Nanayakkara Associate Software Engineer Vetstoria Pvt. Ltd	<i>“Very interesting application because an app like this would be very useful for me daily as I am a person who reads a lot of Sinhala Newspapers. The app generated meaningful summaries in most cases, which is the goal of the solution. As a general user I find this app very useful and efficient. It still has some more updates that can be made but for now it does its job.”</i>
Arkam Ahamed Software Engineer Sysco Labs Pvt. Ltd	<i>“As a Sri Lankan myself I find this application very useful for me as well as for many of the people from the Sri Lankan community. I am a person who scrolls through social media daily. I read a lot of content and instantly forget most of it or get bored by reading them due to their lengthiness. An app like this can save my time as well as help me consume the most important content of a long article. Good work!!”</i>
Mr. Kalaru Nandasena Software Engineer Intern	<i>“I am a hardcore Sinhala meme, document, news, article reader. So, this app would be very useful for me. The current app does generate good summaries even though it still has room for improvements.</i>

DxDy Pvt. Ltd	<i>Personally, I would like to incorporate a question answer system within the app so I will be able to get more info from the content that hasn't been generated within the summary. Making the models and code publicly available also helps future researchers massively."</i>
Mr. Mihin Rathnayake Associate Technical Consultant Enterprise Analytics Pvt. Ltd	<i>"I would say that this is a great research area because I am a person who hates reading long documents as it is a waste of time. An app like this can save a lot of time. As mentioned, even if the model is specific for Sinhala language it still supports other languages too. That is what I want. Great app. I would like to hear about future updates and features too. I would personally like to have it available in the web to use it online"</i>
Mr. Alhan Zuhdi Fahim Undergraduate - SLIIT	<i>"The research area is well considered as it is a problem that is currently faced by a lot of Sri Lankans. Due to the load of data on the web most of the important contents are missed. This application helps the user to grab the most important contents of a long document. The app is very easy to use too. I would like to see some more features in the future. "</i>
Mr. Aqeel Shafy UI/UX Design Intern DxDy Pvt. Ltd	<i>"Personally, I know Hamza as a person who always designs apps with the mindset of making them very customer centric. So, I am not surprised at how well the UI looks. He has achieved the ability to grab user attention by the simple design. Having a light theme was also very good in this case. Making the web app openly available through Github helps a lot with future research."</i>