****

**Web App Backend Engine Powered by NLP Techniques for Flexible Document Summarization**

**Submitted by: Laddie Tan Ji Cheng**

**Supervisor: A/P Chen Lihui  
Examiner: Dr Chan Chee Keong**

School of Electrical & Electronic Engineering

A final year project report presented to the Nanyang Technological University

in partial fulfilment of the requirements of the degree of

Bachelor of Engineering

**2023**

# Table of Contents

[Table of Contents ii](#_Toc131278397)

[Abstract i](#_Toc131278398)

[Acknowledgements ii](#_Toc131278399)

[Acronyms iii](#_Toc131278400)

[List of Figures iv](#_Toc131278401)

[List of Tables v](#_Toc131278402)

[Chapter 1 Introduction 1](#_Toc131278403)

[1.1 Background 1](#_Toc131278404)

[1.2 Objectives and Scope 2](#_Toc131278405)

[1.3 Report Organization 3](#_Toc131278406)

[Chapter 2 Literature Review 4](#_Toc131278407)

[2.1 Front-end Web Development 4](#_Toc131278408)

[2.2 Back-end Web Development 4](#_Toc131278409)

[2.3 Database 5](#_Toc131278410)

[2.4 Document Summarizer 6](#_Toc131278411)

[2.4.1 Extractive Summarization 6](#_Toc131278412)

[2.4.2 Abstractive Summarization 7](#_Toc131278413)

[2.5 Transformer 8](#_Toc131278414)

[2.6 HIWESTSUM 9](#_Toc131278415)

[2.7 Text-to-Text Transfer Transformer 11](#_Toc131278416)

[Chapter 3 Project Design 13](#_Toc131278417)

[3.1 Frameworks and tools 13](#_Toc131278418)

[3.2 Site Map 14](#_Toc131278419)

[3.3 Front-end Design 14](#_Toc131278420)

[3.3.1 Landing Page 14](#_Toc131278421)

[3.3.2 Login and Registration Page 16](#_Toc131278422)

[3.3.3 Profile Page 17](#_Toc131278423)

[3.3.4 Change Credentials Page 18](#_Toc131278424)

[3.3.5 Summarizer Page 18](#_Toc131278425)

[3.3.6 History Page 20](#_Toc131278426)

[3.4 Back-end Design 21](#_Toc131278427)

[3.4.1 Folder Structure 22](#_Toc131278428)

[3.5 Database 23](#_Toc131278429)

[3.5.1 PostgreSQL 23](#_Toc131278430)

[3.5.2 Users Relation 23](#_Toc131278431)

[3.5.3 Texts Relation 24](#_Toc131278432)

[3.5.4 ER Diagram 24](#_Toc131278433)

[3.6 Document Summarizer 25](#_Toc131278434)

[3.6.1 HIWESTSUM 25](#_Toc131278435)

[3.6.2 Text-to-Text Transfer Transformer 28](#_Toc131278436)

[Chapter 4 Implementation 30](#_Toc131278437)

[4.1 Web Application Implementation 30](#_Toc131278438)

[4.1.1 Login and Registration 30](#_Toc131278439)

[4.1.2 Summarizer Page 34](#_Toc131278440)

[4.1.3 Change Credentials Page 34](#_Toc131278441)

[Chapter 5 38](#_Toc131278442)

[Experiments and Results 38](#_Toc131278443)

[5.1 Dataset 38](#_Toc131278444)

[5.1.1 Pre-process Data 39](#_Toc131278445)

[5.2 Experimental Settings 39](#_Toc131278446)

[5.3 ROUGE Evaluation Metrics 40](#_Toc131278447)

[5.4 T5 Experiments 42](#_Toc131278448)

[Chapter 6 43](#_Toc131278449)

[Conclusion and Future Work 43](#_Toc131278450)

[6.1 Conclusion 43](#_Toc131278451)

[6.2 Recommendation in Future Work 44](#_Toc131278452)

[6.2.1 Web application recommendation 44](#_Toc131278453)

[6.2.2 Document Summarizer 45](#_Toc131278454)

[6.3 Learning Reflection 46](#_Toc131278455)

[References 48](#_Toc131278456)

[Appendix (optional) 50](#_Toc131278457)

# Abstract

This final year project concentrated on designing and developing a full-stack web application that provides an efficient and user-friendly platform for summarizing documents. By utilizing natural language processing (NLP) techniques, the web application allows users to summarize lengthy documents into shorter summaries while maintaining the important information.

In this project, front-end technologies consist of HyperText Markup Language (HTML), Cascaded Style Sheet (CSS) and Javascript were utilized to build the user interface and presentation layer of the web application. Back-end technologies consist of Python, Flask and PostgreSQL were used to manage data and process necessary tasks for to support the frontend. Hierarchical Weight Sharing Transformers for Summarization (HIWESTSUM) and Text-to-Text-Transfer Transformer (T5) were utilized for summarizing the documents.

# Acknowledgements

To begin with, I would like to express my sincerest gratitude towards those that had provided me with guidance and assistance which contributed to the accomplishment and success of this project.

In addition, I would like to express my gratitude and thanks to my project supervisor, Associate Professor Chen Lihui for the continuous guidance and attentive supervision provided throughout this project. Moreover, I would also like to extend my gratitude to Xu Jiahao, the PhD in charge, for guiding and providing me with directions for this project.

Last but not least, I would like to express my gratitude and thanks to Mr. Yeo Sung Kheng and MLDA from the School of Electrical and Electronics Engineering for giving me access to the GPU resources and remote server required for this final year project.

Laddie Tan Ji Cheng

March 2023

# Acronyms

|  |  |
| --- | --- |
| ALBERT  API  BERT  BERTSUM  CNN/DM  CSS  C4  DB  DBMS  GPT  GPU  HIWESTSUM  HTML  HTTP  NLP  POSTGRE  REST  ROUGE  T5  WWW | A lite Bidirectional Encoder Representation from Transformers  Application Programming Interface  Bidirectional Encoder Representation from Transformers  BERT for Summarization  Cable News Network/The Daily Mail  Cascade Style Sheet  Colossal Clean Crawled Corpus  Database  Database Management System  Generative Pre-trained Transformer  Graphics Processing Unit  Hierarchical Weight Sharing Transformer for Summarization  HyperText Markup Language  HyperText Transfer Protocol  Natural Language Processing  PostgreSQL  Representational State Transfer  Recall-Oriented Understudy for Gisting Evaluation  Text-to-Text Transfer Transformers  World Wide Web |

# List of Figures

[Figure 1 Transformer Architecture [14] 9](file:///C:\Users\Laddi\Desktop\FYP%20REPORT\FYP_Report_Template.docx#_Toc131278458)

[Figure 2 HIWESTSUM architecture [6] 10](#_Toc131278459)

[Figure 3 - A Text-to-Text Framework 11](file:///C:\Users\Laddi\Desktop\FYP%20REPORT\FYP_Report_Template.docx#_Toc131278460)

[Figure 4 - Site Map of the project 14](file:///C:\Users\Laddi\Desktop\FYP%20REPORT\FYP_Report_Template.docx#_Toc131278461)

[Figure 5 – Main Landing Page 15](file:///C:\Users\Laddi\Desktop\FYP%20REPORT\FYP_Report_Template.docx#_Toc131278462)

[Figure 6 – Extractive Summarization Knowledge Page 15](file:///C:\Users\Laddi\Desktop\FYP%20REPORT\FYP_Report_Template.docx#_Toc131278463)

[Figure 7 - Abstractive Summarization Knowledge Page 16](file:///C:\Users\Laddi\Desktop\FYP%20REPORT\FYP_Report_Template.docx#_Toc131278464)

[Figure 8 – Email sent to user after registration 16](file:///C:\Users\Laddi\Desktop\FYP%20REPORT\FYP_Report_Template.docx#_Toc131278465)

[Figure 9 - Login and Registration Page 17](file:///C:\Users\Laddi\Desktop\FYP%20REPORT\FYP_Report_Template.docx#_Toc131278466)

[Figure 10 - Main Profile Page 17](file:///C:\Users\Laddi\Desktop\FYP%20REPORT\FYP_Report_Template.docx#_Toc131278467)

[Figure 11 - Changing Credentials Page 18](file:///C:\Users\Laddi\Desktop\FYP%20REPORT\FYP_Report_Template.docx#_Toc131278468)

[Figure 12 - Summarizer Page with result 19](file:///C:\Users\Laddi\Desktop\FYP%20REPORT\FYP_Report_Template.docx#_Toc131278469)

[Figure 13 - Summarizer Page without result. 19](file:///C:\Users\Laddi\Desktop\FYP%20REPORT\FYP_Report_Template.docx#_Toc131278470)

[Figure 14 - History Page with Result 20](file:///C:\Users\Laddi\Desktop\FYP%20REPORT\FYP_Report_Template.docx#_Toc131278471)

[Figure 15 – Empty History Page 21](file:///C:\Users\Laddi\Desktop\FYP%20REPORT\FYP_Report_Template.docx#_Toc131278472)

[Figure 16 - Folder Structure of this Project 22](file:///C:\Users\Laddi\Desktop\FYP%20REPORT\FYP_Report_Template.docx#_Toc131278473)

[Figure 17 - users relation 23](file:///C:\Users\Laddi\Desktop\FYP%20REPORT\FYP_Report_Template.docx#_Toc131278474)

[Figure 18 - texts relation 24](file:///C:\Users\Laddi\Desktop\FYP%20REPORT\FYP_Report_Template.docx#_Toc131278475)

[Figure 19 - ER diagram 24](file:///C:\Users\Laddi\Desktop\FYP%20REPORT\FYP_Report_Template.docx#_Toc131278476)

[Figure 20 - Overall Architecture for Summarization 29](#_Toc131278477)

[Figure 21 - Error Message 1 30](file:///C:\Users\Laddi\Desktop\FYP%20REPORT\FYP_Report_Template.docx#_Toc131278478)

[Figure 22 - Error Message 2 31](file:///C:\Users\Laddi\Desktop\FYP%20REPORT\FYP_Report_Template.docx#_Toc131278479)

[Figure 23 - Registration Error Message 1 32](file:///C:\Users\Laddi\Desktop\FYP%20REPORT\FYP_Report_Template.docx#_Toc131278480)

[Figure 24 - Registration Error Message 2 32](file:///C:\Users\Laddi\Desktop\FYP%20REPORT\FYP_Report_Template.docx#_Toc131278481)

[Figure 25 - Registration Error Message 3 33](file:///C:\Users\Laddi\Desktop\FYP%20REPORT\FYP_Report_Template.docx#_Toc131278482)

[Figure 26 - Registration Error Message 4 33](file:///C:\Users\Laddi\Desktop\FYP%20REPORT\FYP_Report_Template.docx#_Toc131278483)

[Figure 27 - Summarizer Error Message 34](file:///C:\Users\Laddi\Desktop\FYP%20REPORT\FYP_Report_Template.docx#_Toc131278484)

[Figure 28 - Page to change credentials 35](file:///C:\Users\Laddi\Desktop\FYP%20REPORT\FYP_Report_Template.docx#_Toc131278485)

[Figure 29 - Changing Email Error Message 1 35](file:///C:\Users\Laddi\Desktop\FYP%20REPORT\FYP_Report_Template.docx#_Toc131278486)

[Figure 30 - Changing Email Error Message 2 36](file:///C:\Users\Laddi\Desktop\FYP%20REPORT\FYP_Report_Template.docx#_Toc131278487)

[Figure 31 - Changing Email Error Message 36](file:///C:\Users\Laddi\Desktop\FYP%20REPORT\FYP_Report_Template.docx#_Toc131278488)

[Figure 32 - Changing Name Error Message 37](file:///C:\Users\Laddi\Desktop\FYP%20REPORT\FYP_Report_Template.docx#_Toc131278489)

[Figure 33 - Changing Password Error Message 37](file:///C:\Users\Laddi\Desktop\FYP%20REPORT\FYP_Report_Template.docx#_Toc131278490)

[Figure 34 - CNN/DM Raw Example (id: 42c027e4ff9730fbb3de84c1af0d2c506e41c3e4) 38](file:///C:\Users\Laddi\Desktop\FYP%20REPORT\FYP_Report_Template.docx#_Toc131278491)

# List of Tables

[Table 1 - Sorted HIWESTSUM Output 26](#_Toc131278492)

[Table 2 - Cumulated Probabilities 27](#_Toc131278493)

[Table 3 - Top K sentences based on threshold. 28](#_Toc131278494)

[Table 4 - T5 Rough Score 42](#_Toc131278495)

# Introduction

## Background

Natural language processing (NLP), a branch of artificial intelligence, enables computers to interact with and comprehend human language. It has been a rapidly growing field in recent years, with the market revenue increasing and applications extending to various industries in healthcare, finance and social media. Additionally, due to the widespread use of social media and the internet, the increase in data has allowed researchers to develop models such as ChatGPT[1], GPT3[2], BERT[3] and T5[4] that have revolutionized the field of NLP and opened more opportunities for artificial intelligence to comprehend and generate human languages.

With the increasing consumption of technologies and online articles, the demand for document summarization will be expanding. Document summarization is the process of breaking down lengthy articles into digestible paragraphs or sentences while preserving important information and creating summaries that are coherent, non-redundant and readable. [5]

The project will be using the model known as Hierarchical Weight Sharing Transformers for Summarization (HIWESTSUM) developed by a former student from the School of Electrical and Electronics Engineering. HIWESTSUM has shown that being trained on a smaller dataset may nevertheless produce comparable results to state-of-the-art models. The HIWESTSUM explored the use of hierarchical structures for document representation by encoding the document from word level to sentences and document level representation. A score will then be provided for the sentences. [5] Additionally, the HIWESTSUM model adopts a weight sharing mechanism that can reduce the number of parameters in the architecture significantly which is almost ten times lesser than the currently used models for extractive summarization known as fine-tune BERT. [3]

On the other hand, a web application is an application program consisting of client-side and server-side components that can be stored on a remote server and sent over the internet through mobile applications and web browsers such as Google Chrome and Mozilla Firefox. They are used for numerous purposes such as online shopping and online banking, but in recent years, web applications have evolved to provide more sophisticated features and functionalities like real-time communication, data visualization and machine learning capabilities.

## Objectives and Scope

The objective of this project is to create a web application with a suitable document summarization technique that allows users to produce a concise summary that is flexible in length from their long articles. The objective is broken down into two parts that are functions of the web application and the document summarizer shown in the following list:

1. Functions of Web Application
   1. User
      1. Register accounts for new users.
      2. Login for current user.
      3. Logging out.
   2. Profile Page
      1. View user profile.
      2. Update credentials.
   3. Summarization Page
      1. Navigation bar.
      2. Summarizing documents via file upload.
   4. History
      1. Table to view user’s past summaries.
2. Summarizer
   1. Fine-tune a T5 model for abstractive summarization.
   2. Research on T5 model and the difference between baseline and fine-tune model.

## Report Organization

This report consist of five chapters and the structure of this report consists of the following organization.

1. Chapter 1: Introduction

This chapter give a brief introduction and objectives of this final year project.

1. Chapter 2: Literature Review

The following chapter provide background knowledge and concepts on the technologies used.

1. Chapter 3: Project Design

Overview of how the web application is designed and developed with the summarization model.

1. Chapter 4: Implementation and Result

This chapter illustrated the details and features available and show the difference between a baselines and fine-tuned model.

1. Chapter 5: Experiment and Result

This chapter covers the difference between T5 (‘t5-base’) and T5 (‘fine-tuned’) model.

1. Chapter 6: Conclusion and Future Work

Covers the conclusion of this project with suggestion for future development for this project.

# Literature Review

This chapter shows the research on various applications, technologies, programming stacks and engineering tools that the author has utilized for the development of the web application and summarization model.

## Front-end Web Development

Running on client-side, front-end development is the process of creating a user interface and experience of a website or web application. It often includes the designing and coding of the web application that users interact with and is responsible for the creation of user-facing elements.

Three programming languages were used in the development of this final year project, and they are HyperText Markup Language (HTML), Cascading Style Sheets (CSS) and Javascript. To begin with, HTML is a language that is being used in most sites. The primary function of HTML is to structure the web page and its content. [7] Some examples include headings, retrieving online information via hypertext links and including spreadsheet, video clip and sound clip. In addition, CSS is a stylesheet language used for styling, layout of web pages and describing how elements should be rendered on the screen. [8] Finally, Javascript is designed to work with HTML and CSS, often used to add functionality to web pages, such as form validation, interactive maps and dynamic content without the need for page refresh. They lay the foundations for web pages and render web pages served by servers.

## Back-end Web Development

Backend web development is another subset of web development that runs on the server-side and powers the website or web application by handling the logic and data processing. The popular backend languages or technologies are Django, Flask, Ruby on Rails and Express.js. Backend developers often work with the frontend developers that operate the client-side of web development. Two sides collaborate to create a web application that provides a good and clean user interface and experience.

Additionally, another role of backend developers is to maintain existing web applications, making sure that the application is optimized, efficient and smooth for the users. Also, they are responsible for fixing any bugs and resolving issues that users or product managers encounter while using the application. Overall, backend developers are an essential part of web development because they power and ensure that web applications are running smoothly.

## Database

A well-organized group of data that is related to each other in a meaningful way and that can be accessed in different logical orders is called a database. Organizations often have a huge volume of data. This data will be accessed by numerous users and the need for long-term storage is required so database is usually used by them. [9] Database systems are either housed on server, data center or virtually on cloud infrastructure or cloud databases. An example database for an eCommerce app will include customer, business and relationship data. There are multiple types of databases and some of the most used ones are relational, non-relational, graph and object- oriented databases. The primary focus for this final year project will be relational databases.

A formally described table that contains an organized collection of data items is known as a relational database which allows data to be reassembled and accessed in multiple different ways. A set of tables in relational database are referred to as relation which contains columns known as attributes. Each row in the relation is a unique instance of data. When creating a relational database domain, constraints are applied to the data along with possible values. The relation between tables makes it a ‘relation’ table and a few assumptions are needed to extract data from the database resulting in the same database being viewed in many ways. [10] Structured query language (SQL) is the primary interface to communicate with relational databases. The common types of relational database management systems (RDMS) are MySQL, MS Access, Oracle and PostgreSQL and they use SQL as their standard database language.

## Document Summarizer

Document summarizing is the process of identifying and extracting key sentences from user’s input documents to produce a summary. Moreover, document summarization produces summaries that are shorter than the input document and maintain the integrity of the information content and overall meaning. [11] As compared to human-generated summaries, artificial intelligence generated summaries are less biased. By using a summarizer, it helps to reduce reading time.

Extractive and abstractive summarization are the two types of summarization tasks. The main idea of extractive summarization is to extract important and isolate pre-existing sentences or paragraphs from the document and concentrate them together to form a summary. On the other hand, the task of the abstractive summarization method is to generate a short and concise summary while capturing the salient ideas of the document. The abstractive summaries often contain phrases or sentences that are not in the original document. In this project, both methods will be used for the summarization to generate a short summary for the user.

### Extractive Summarization

In extractive summarization, important sentences are selected considering the linguistic and statistical features of the words in the original document. The objective of this summarizer is to generate a coherent final summary from the extracted important content and the length of the summary depends on the compression rate. These methods assign saliency scores to sentences in the documents and those highly scored sentences will be used to form a summary.

Extractive summarization often adopts a statistical approach by using an intermediate representation acquired from capturing the statistical features of text and assigning scores to sentences based on the representation. The sentences that are highly rated will be selected and combined to form the summary. [12] Some features include the position of sentences, positive and negative keywords, centrality of sentences and the resemblance of sentences to the title.

This method of summarizing is independent of language. Hence, any linguistic knowledge or complex linguistic processing is not required and allowing the summarizer to summarize text in any language.

### Abstractive Summarization

Instead of extracting important sentences from the document, abstractive summarization adopts another approach where the summary is generated by either rephrasing or using new words that are not from the original document. Semantic information regarding the text is analyzed and internal representation of the text. New sentences are generated out of the original text using deep analysis and reasoning as required. [13]

Abstractive summarization adopts the deep learning and neural network approach, where encoders and decoders are used and by putting them together will result in a combined neural network. The encoder captures the context of the document by converting the text of the document into vector representation. Based on previous words, the decoder assists in finding the next work in the summary. The forms of input and output are usually in sequence which makes the abstractive method a sequence-to-sequence learning problem.

The process of abstractive summarization requires extensive and advanced natural language processing. Hence, this results in complexity and is probably more challenging as compared to extractive summarization.

## Transformer

A network that employs an encoder and decoder architecture known as Transformer [14] was introduced in 2017. Transformers can process the text and determine the word embedding simultaneously, unlike the recurrence neural network (RNN) which processes the word and generates the word embedding one time step at a time.

Additionally, Transformers includes vector encoding of the word position in a sentence which allows the word embedding to contain positional information. Another feature of the Transformer is known as attention. For every word in the sentence, an attention vector is generated in the attention block which captures the contextual relationships between the other words in a sentence.

The overall architecture of the Transformer shown in figure 1consists of two parts the encoder and the decoder. In this project, HIWESTSUM model that leverage on transformers encoder that are in existing pretrained transformer-based model such as BERT and ALBERT. [6]

This literature review will only focus on encoder structure because the structure of HIWESTSUM uses representations from encoders for computation of sentence scores. The encoder consists of a stack with N = 6 identical layers with two sub-layers on each layer. Multi-head self-attention mechanisms and a simple, position-wise fully connected feed-forward network (FFN) make up the first and second layer. [14] Each sub-layer output is LayerNorm(x + sublayer(x)) where sublayer(x) is the function that the sub-layer implements.

## Diagram Description automatically generatedHIWESTSUM

Figure 1 Transformer Architecture [14]

HIWESTSUM model was trained with the CNN/DM dataset using supervised and extractive objectives. The dataset provides a gold summary for each article that can be used as ground truth. The HIWESTSUM was built on top of BERTSUM, which resulted in the architecture of HIWESTSUM being similar to BERTSUM. The HIWESTSUM model employs hierarchical structure to encode word tokens into sentence-level representations, followed by encoding the sentence-level representation into document level representations to score the sentences. As shown in figure 2 below, the model use document level representation of hierarchical representations achieves extractive summarization by computing the score for individual sentences in the document.[2]

Diagram

Description automatically generated

Figure 2 HIWESTSUM architecture [6]

Additionally, the model uses transformer from the same ALBERT that was utilized in the previous sentence representation learning. [15] Instead of the ones from the extractive layers. By adopting this approach, the performance may improve and the parameters will stabilize due to the sharing and attention between different hierarchical levels. As compared to ALBERT (‘albert-based-v2’) and BERT (‘bert-base-uncased’), HIWESTSUM achieved comparable results with a lighter and faster model that only utilized 11.69 million with ALBERT. [2]

Furthermore, the model utilized all hierarchical level representation within the document. Preprocessing the input sequence of the document by adding special tokens “[CLS]” and [“SEP”]. [“CLS”] is used for classification during training and is added to the start of each sentence. On the other hand, [“SEP”] is used to distinguish the sentences and is added after each sentence. The “[CLS]” output vector is utilized to aggregate features of the respective sentence and passed to the ALBERT model as input. The sentence-level representation produced by ALBERT will be fed into the same encoder stack that was used previously to acquire the document level representation.

## Text-to-Text Transfer Transformer

Recent advances in NLP have shown that transfer learning, a method where a pre-trained model that was trained on a generic task is then used to fine-tune on specific downstream tasks such as summarization, classification and machine translation. It allows the model to converge faster and has a lower requirement for fine-tuning data. [15] The inspiration for fine-tuning is based on how humans acquire knowledge. Instead of working on the problem from scratch, existing knowledge is relied on and built upon it, especially when the problem is related. Transfer learning is typically used when the training dataset is not large enough for certain applications.

Diagram

Description automatically generatedIn this project, the Text-to-Text Transfer Transformer (T5) model is utilized for the abstractive summarization portion. T5 is essentially a survey on how modern transfer learning methods are used in language understanding. Hence, the researchers propose a unified framework that attempts to combine all language problems into a text-to-text format as shown in figure 3. With the unified approach, effective comparison can be achieved between different transfer learning objectives. At the same time, exploring the limits of transfer learning for NLP. [16] The T5 model is similar to the original Transformer [14] architecture except for the removal of Layer Norm bias. The placement of normalization will be moved out of the residual path and using a different position embedding scheme. [16]

Figure 3 - A Text-to-Text Framework

The T5 model leverages Common Crawl which was used as a source of text data for NLP and scraped from the web that produces around 20TB of scraped data each month. Some heuristics are used to clean up the Common Crawl text because the majority of the text is not in natural language. Language detection is used to detect English only since most of the downstream tasks are focused on English. After cleaning and filtering, the resultant dataset used for training is known as the Colossal Clean Crawled Corpus (C4) consisting of reasonably clean and natural English text and about 750 GB used during pre-training.

The main purpose of this paper is to measure general language learning abilities and the performance of different downstream tasks with their respective benchmarks, including machine translation, abstractive summarization, and text classification but this project will only focus on abstractive summarization.

Since this is a text-to-text format, the input will be a text that is fed to the model for context or conditioning and then produces some output text. For the specification of downstream tasks, a task-specific prefix which comes in the form of text is added to the original input sequence before passing it to the model. As shown in figure 3 above, ‘summarize:’ prefix is added to perform summarization.

# Project Design

## Frameworks and tools

This section provides justification for the framework, libraries and tools used.

For designing and building responsive web applications, HTML, CSS, Javascript and Bootstrap were utilized. Bootstrap is a free and open-source popular CSS framework for developing and creating websites or web pages. Designed to enable responsive development of websites by providing collections of syntax for template designs. Hence, it is efficient to build websites without worrying about the basic commands and functions.

Python was utilized to run the summarization and backend web framework. For summarization, it allows quick integration with the HIWESTSUM model which was trained in Python. Additionally, for fine-tuning of Text-to-Text Transfer Transformers requires the library called HuggingFace on Python. HuggingFace is an open-source and publicly available library that specializes in NLP technologies and the company is well-known for developing and maintaining several popular datasets, tokenizers and models. The Flask web framework was used for the backend of the web application. Flask is a lightweight web framework for building web applications in Python.

The database is used to store user information for all accounts, as well as original document text and summarized text. The technology stack for the database used in this project is PostgreSQL.

## Site Map

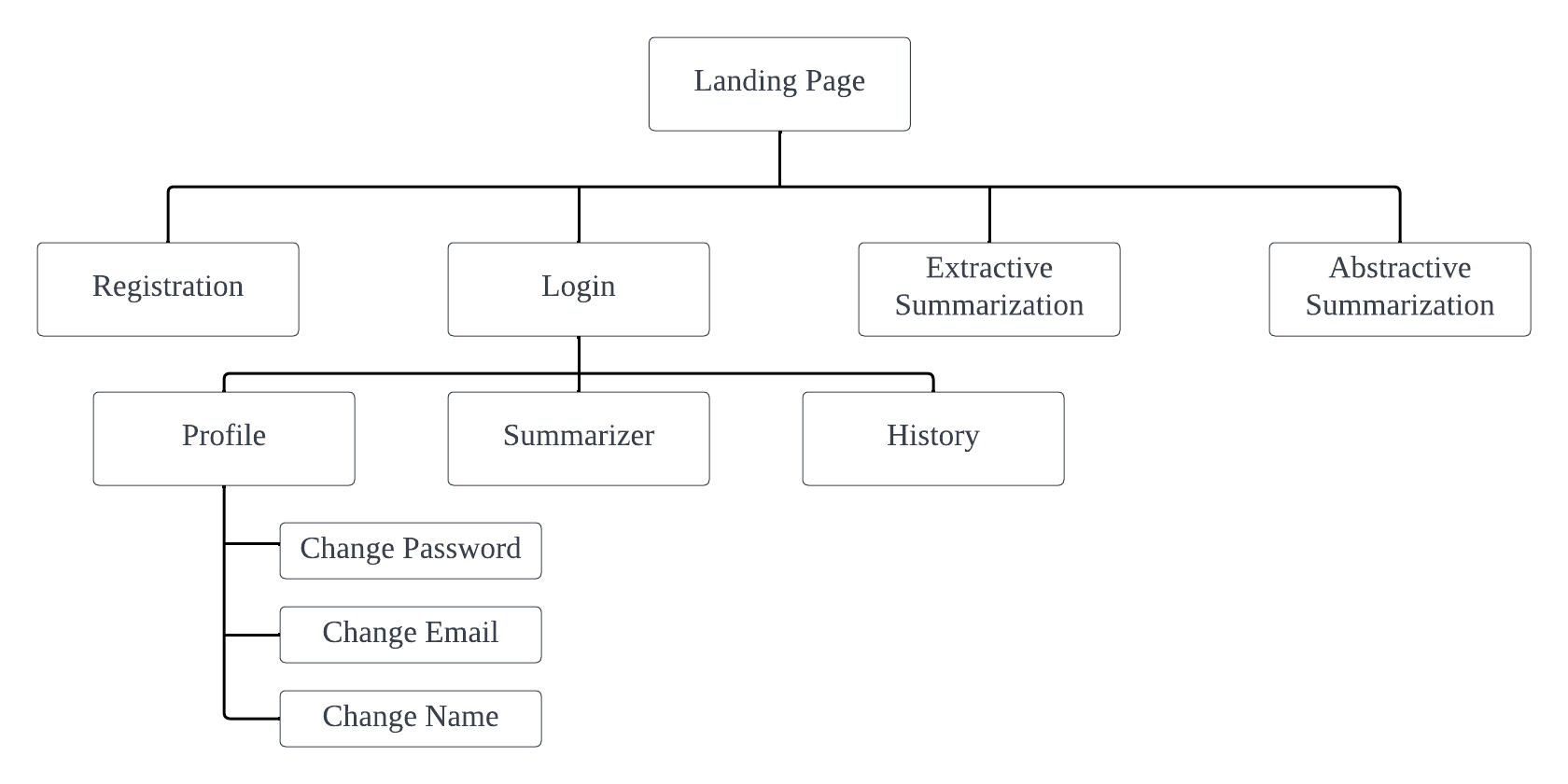
A site map is usually created before the development of the project, it helps the developer plan for the site and gives a bird’s-eye view of the whole architecture. This project aims to provide users with a document summarizer that will capture the original document and summary into the database, allowing the user to utilize them in the future. Users can register for an account using their email address and gain access to the features as shown in the site map in figure 4.

Figure 4 - Site Map of the project

## Front-end Design

The primary pages and features accessible for users are as follows:

1. Landing Page
2. Profile Page
3. Summarizer Page
4. History Page

### Landing Page

Every user that enters the website will be directed to the landing page. The main landing page, as shown in figure 5, shows the background information on document summarization and the types of summarizations. The button on the bottom redirects the user to the specific summarization technique and the page layout for the two summarization techniques is shown in figure 6 (extractive summarization) and figure 7 (abstractive summarization).

Graphical user interface

Description automatically generatedDiagram

Description automatically generated with low confidence

Figure 5 – Main Landing Page

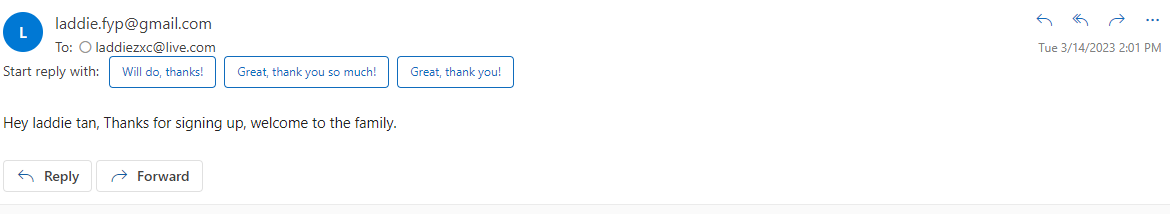
Figure 6 – Extractive Summarization Knowledge Page

### Text Description automatically generated with medium confidenceLogin and Registration Page

Figure 7 - Abstractive Summarization Knowledge Page

These pages allow the new user to register for accounts and login for existing users. On the right of figure 9 is the design of the user registration page. Upon successful account creation, the user will be redirected to the login page shown in the left panel of figure 9 and an email will be sent to the user’s email that was used to create the account. The template of the email is shown in figure 8.

Figure 8 – Email sent to user after registration



### Graphical user interface, application Description automatically generatedGraphical user interface, application Description automatically generatedProfile Page

Figure 9 - Login and Registration Page

This page shows the user profile, name and email of the user are shown, and the user can change their name, email and password on this page. The main profile page is shown in figure 10 below.

Graphical user interface, application

Description automatically generated

Figure 10 - Main Profile Page

### Change Credentials Page

Users are allowed to change their credentials and the requirement to change the credentials is that the user must enter their password. As shown in figure 11, the figure shows the page for changing passwords, but the design is the same for changing other credentials like email and names.

Graphical user interface

Description automatically generated

Figure 11 - Changing Credentials Page

### Summarizer Page

This summarizer page allows the user to upload a file that is either in PDF or TXT format and the cumulative distribution allows the user to select the number of sentences to return based on the HIWESTSUM extractive summarization method. In figure 12, the original text is located on the left, extractive summary produced by HIWESTSUM located in the middle and the abstractive summary produced by T5 (‘fine-tuned’) model on the right. The extracted sentences from extractive summary will be highlighted to allow the user to know where the highly scored sentences are in the original text. An empty summarizer page is shown in figure 13.

Graphical user interface, text, application

Description automatically generated

Figure 12 - Summarizer Page with result

Graphical user interface, application

Description automatically generated

Figure 13 - Summarizer Page without result.

### History Page

Text

Description automatically generatedThis page accesses the database to locate all previous documents and summary text for the current user, and all the result from the database will be shown on this page. As shown in figure 14, the result of user’s previously summarized document will be shown on the page in a table. The columns will be like the table used in the database with a cleaner text. Pagination is added if the number of rows exceeds five per table and a search bar for the user to search for the document they are looking for. Users can also sort the result based on individual columns in the table. For new users without any history, an empty page will show in figure 15.

Figure 14 - History Page with Result

Graphical user interface

Description automatically generated

Figure 15 – Empty History Page

## Back-end Design

Backend built using Python Flask framework, with HyperText Transfer Protocol (HTTP) request and response architecture. HTTP is commonly used to communicate over the World Wide Web (WWW). A request generated by the client will be sent over to the server in the form of a request method with the protocol version and every request message will contain a method that will tell the server what action to perform. The common HTTP methods are GET, POST, PUT and DELETE but in this project only GET and POST were utilized. [17]

### Text Description automatically generated with medium confidenceFolder Structure

Figure 16 - Folder Structure of this Project

* HIWESTSUM checkpoint: This folder contains the trained HIWESTSUM model for extractive summarization.
* T5 checkpoint: contain the fine-tuned model for the Text-to-Text Transfer Transformer (T5) model.
* Models, node modules and src: contain Python files that run the T5 and HIWESTSUM models for summarization.
* Static: Contain static files like images, CSS and JavaScript files.
* Templates: A Folder containing HTML files for the front-end.
* PY files: Backend files that handle server-side tasks.

Hence, with the HTTP method implemented on the back-end of this project, the front-end or client-side can make HTTP requests to create, read, update or delete data or resources from the database in Postgres.

## Database

### PostgreSQL

Postgres is a powerful, open-source, object-relational database that supports both relational and non-relational databases and it is a highly stable database management system. Being one of the most advanced database servers available and was developed by a group of open-source software proponents. [18]

Using Postgres allows simple integration with Flask web framework via Python libraries such as SQLAlchemy and Psycopg2. It is a default database choice for many Python developers due to the features that make it robust and stable when compared to other databases.

### Users Relation

Graphical user interface, application

Description automatically generatedUser relation is used to store data like a user’s credentials to facilitate authentication and other profile related details. The attributes are user ID as the primary key, email, name and password.

Figure 17 - users relation

### Texts Relation

Texts relation stores the user’s original document, the extractive summary result, the abstractive summary result, text ID as the primary key and user ID the foreign key.

### Graphical user interface, application Description automatically generated with medium confidenceER Diagram

Figure 18 - texts relation

A picture containing graphical user interface

Description automatically generatedThe ER diagram is a structural design of the database and figure 19shows the relationship between users relation and texts relation. The relationship type is a one-to-many relationship where one user can have multiple texts, whereas one text only belongs to one user.

Figure 19 - ER diagram

## Document Summarizer

In this project, both summarization techniques were explored and implemented on the web application. The extractive portion uses the HIWESTSUM model to extract sentences that are of high importance to the document. Consequently, the output of HIWESTSUM will be passed onto the T5 (‘fine-tuned’) model to produce an abstractive summary.

### HIWESTSUM

Being the first model to process the original text, the HIWESTUM model returns the sentences that are highly scored in a table. Instead of using the scores produced by the model, SoftMax function was used to turn the scores into probabilities for every sentence and the sum of all probabilities will add up to one. Subsequently, the sentences will be sorted in descending order based on the probabilities. Table 1is an example after running the HIWESTSUM model on a document and HIWESTSUM scores through the SoftMax function.

|  |  |  |
| --- | --- | --- |
| Sentences | HIWESTSUM Score | Probability |
| British hurdler Sarah Claxton is confident she can win her first major medal at next month's European Indoor Championships in Madrid. | 1.614644 | 0.137048 |
| The 25-year-old has already smashed the British record over 60m hurdles twice this season, setting a new mark of 7.96 seconds to win the AAAs title. | 1.279528 | 0.098024 |
| "I am quite confident," said Claxton. | 1.190752 | 0.089697 |
| "But I take each race as it comes. | 1.201512 | 0.090667 |
| "As long as I keep up my training but not do too much, I think there is a chance of a medal." | 1.15594 | 0.086628 |
| Now, the Scotland-born athlete owns the equal fifth-fastest time in the world this year. | 1.149938 | 0.08611 |
| Claxton has won the national 60m hurdles title for the past three years but has struggled to translate her domestic success to the international stage. | 1.137436 | 0.08504 |
| And at last week's Birmingham Grand Prix, Claxton left European medal favourite Russian Irina Shevchenko trailing in sixth spot. | 1.123201 | 0.083838 |
| For the first time, Claxton has only been preparing for a campaign over the hurdles - which could explain her leap in form. | 1.09748 | 0.081709 |
| In previous seasons, the 25-year-old also contested the long jump but since moving from Colchester to London she has re-focused her attentions. | 1.083742 | 0.080594 |
| Claxton will see if her new training regime pays dividends at the European Indoors which take place on 5-6 March. | 1.084357 | 0.080644 |

Table 1 - Sorted HIWESTSUM Output

Thereafter, the project will utilize the cumulative distribution function (CDF) which is the probability that a random variable will take a value less than or equal to the random variable. The purpose of CDF is to compute the cumulated probability in every sentence. By doing so, the user can control the cumulated probability and the result will return the top K sentences with the cumulated probability lesser than user’s input threshold. In table 2, the table shows the cumulated probability after passing through the function for calculation. In table 3**,** the result returns the sentences with the cumulated probability less than or equal to the user’s input threshold. In this context, assume that the user’s threshold is 0.5.

|  |  |  |
| --- | --- | --- |
| Sentences | Probability | Cumulated Probability |
| British hurdler Sarah Claxton is confident she can win her first major medal at next month's European Indoor Championships in Madrid. | 0.137048 | 0.137048 |
| The 25-year-old has already smashed the British record over 60m hurdles twice this season, setting a new mark of 7.96 seconds to win the AAAs title. | 0.098024 | 0.235072 |
| "I am quite confident," said Claxton. | 0.089697 | 0.325739 |
| "But I take each race as it comes. | 0.090667 | 0.415436 |
| "As long as I keep up my training but not do too much, I think there is a chance of a medal." | 0.086628 | 0.502065 |
| Now, the Scotland-born athlete owns the equal fifth-fastest time in the world this year. | 0.08611 | 0.588166 |
| Claxton has won the national 60m hurdles title for the past three years but has struggled to translate her domestic success to the international stage. | 0.08504 | 0.673206 |
| And at last week's Birmingham Grand Prix, Claxton left European medal favourite Russian Irina Shevchenko trailing in sixth spot. | 0.083838 | 0.757053 |
| For the first time, Claxton has only been preparing for a campaign over the hurdles - which could explain her leap in form. | 0.081709 | 0.838762 |
| In previous seasons, the 25-year-old also contested the long jump but since moving from Colchester to London she has re-focused her attentions. | 0.080594 | 0.919406 |
| Claxton will see if her new training regime pays dividends at the European Indoors which take place on 5-6 March. | 0.080644 | 1.000000 |

Table 2 - Cumulated Probabilities

|  |  |  |
| --- | --- | --- |
| Sentences | Probability | Cumulated Probability |
| British hurdler Sarah Claxton is confident she can win her first major medal at next month's European Indoor Championships in Madrid. | 0.137048 | 0.137048 |
| The 25-year-old has already smashed the British record over 60m hurdles twice this season, setting a new mark of 7.96 seconds to win the AAAs title. | 0.098024 | 0.235072 |
| "I am quite confident," said Claxton. | 0.089697 | 0.325739 |
| "But I take each race as it comes. | 0.090667 | 0.415436 |

Table 3 - Top K sentences based on threshold.

### Text-to-Text Transfer Transformer

In this project, the next focus is to explore the use of abstractive summarization. Hence, the Text-to-Text Transfer Transformer (T5) model is proposed for the abstractive summarization with the aim to fine-tune the model for downstream tasks specifically summarization. The main two directions of document summarization are: (i) to explore the base T5 model and (ii) to fine-tune the model for downstream tasks. CNN/DM dataset was used to fine-tune the model for summarization because the dataset provides gold summaries for every document which will be used as ground truth.

The input for the T5 model will be the output from the HIWESTSUM model. The output from the HIWESTSUM model will be concatenated together to form a paragraph and fed into the T5 model for summarization. Figure 20 below shows the overall process for the document summarization portion. Both outputs will be shown to the user on the summarizer page. The bounding box will be what the users see on the summarization page.

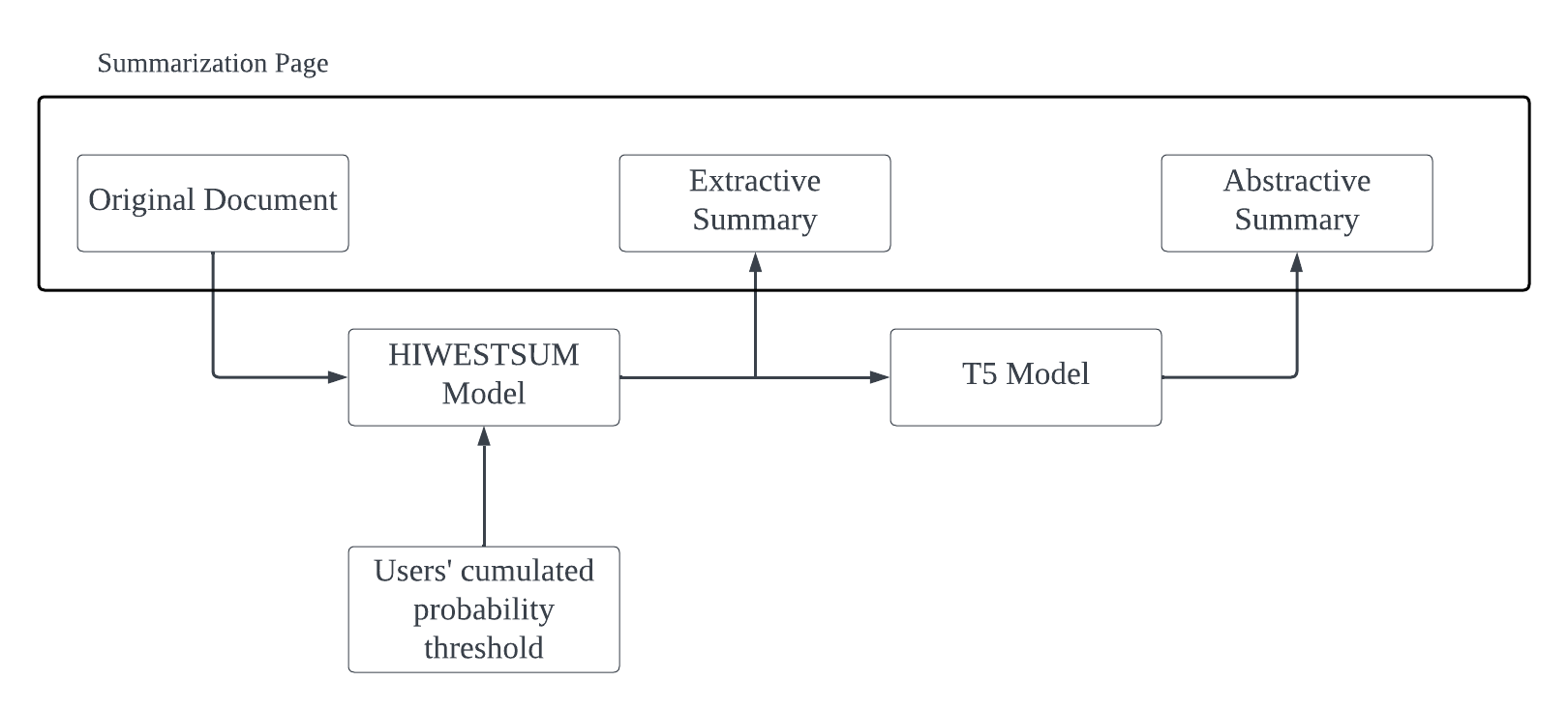


Figure 20 - Overall Architecture for Summarization

# Implementation

This chapter discusses the implementation details of the web application with both front-end and back-end components. There will be described use-case scenario, the implementation details will be documented with a workflow diagram or shown in figures and code samples throughout the chapter. On the other hand, the dataset used for fine-tuning will be analyzed and comparison between T5 (‘t5-base’) and T5 (‘fine-tuned’) for summarization will be compared.

## Web Application Implementation

### Login and Registration

Details such as email and password from the form will send a POST request to the backend, the backend will then search the database for a match.

**Login Page**

1. Graphical user interface

   Description automatically generatedEmail and password are retrieved and match with the user’s relation in Postgres. If no email is found in the database, an error message will be shown to the user on the login page as shown in figure 21.

Figure 21 - Login Error Message 1

1. Email is first queried from the database followed by password and if the password does not match, an error message will be shown to the user on the login page as shown in figure 22.

Graphical user interface

Description automatically generated

Figure 22 – Login Error Message 2

1. If email is found and password matched, the user will be redirected to the summarizer page as shown in figure 13.

**Registration Page**

Credentials such as name, password and email are required on the registration page. Additionally, there will be an extra text box for passwords to make sure that the user enters the correct desired password.

1. Graphical user interface, application

   Description automatically generatedEmail will be queried first and if a match is found in the database, an error will be thrown with the message “email already exist” on the registration page as shown in figure23 below.

Figure 23 - Registration Error Message 1

1. An error will be prompted to the user if the user’ name contains any special characters. The error message is shown in figure 24below.

Graphical user interface, text, application

Description automatically generated

Figure 24 - Registration Error Message 2

1. Graphical user interface, text, application

   Description automatically generatedThe first requirement for password is eight characters and any password less than eight characters will prompt an error message as shown in figure 25**.** The second requirement is that the password from the two textboxes need to match or else an error message shown in figure 26will be shown to the user.

Figure 25 - Registration Error Message 3

Graphical user interface, application

Description automatically generated

Figure 26 - Registration Error Message 4

1. Successful Registration will direct the user to the login page and an email will be sent to the users’ email.

### Summarizer Page

Graphical user interface, application

Description automatically generatedGraphical user interface, application

Description automatically generatedTo successfully summarize a document, two requirements must be fulfilled. They are: (i) Be in TXT or PDF format and (ii) have token count less than 512. The first requirement was implemented because more of the text files are in the two formats mentioned. The latter is because of the structure of BERT which only allows max token of 512. To count the number of tokens, the Natural Language ToolKit (NLTK) library was utilized. The left of figure 27shows the error message when the wrong file format is submitted and right of figure 27 prompts the user an error message if the tokens in the document exceed 512.

Figure 27 - Summarizer Error Message

### Change Credentials Page

This page allows users to change their credentials such as name, email and password but to successfully change the credentials, users must be signed in. For every credential, the requirement will be to enter the password. If the user successfully changed any of the credentials, the database will be updated.

Graphical user interface, text, application

Description automatically generatedFor every case, if the users’ input password does not match the database password, an error message will be shown to the user. Figure 28 shows the error message when password user entered does not match with the database password when changing user’s name. The same error message will be displayed for other credentials when the password does not match.

Figure 28 - Page to change credentials

**Change Email**

1. Graphical user interface, text, application

   Description automatically generatedTo change the email, the user must enter the password and if the password does not match. An error message will be shown to the user as shown in figure 29.

Figure 29 - Changing Email Error Message 1

1. Graphical user interface, text, application

   Description automatically generatedAn error message will be shown to the user if the new email is similar to the old one or exists in the database. As shown in figure 30, the error message is shown below the email textbox.

Figure 30 - Changing Email Error Message 2

1. Graphical user interface, text, application

   Description automatically generatedThe email entered by the user must be in the correct format. Otherwise, an error message will be shown to the user. As shown in figure 31, the error message will appear when user tries to submit with an invalid email format.

Figure 31 - Changing Email Error Message

**Change Name**

Graphical user interface, text, application, chat or text message

Description automatically generatedAn error message will be shown if the textbox for changing users’ names contains any special characters. As shown in figure 32, the page will be refreshed and with the message. If the change is successful, the name will be updated on the users relation in the database and the updated name will be reflected on the profile page.

Figure 32 - Changing Name Error Message

**Change Password**

Graphical user interface, text, application

Description automatically generatedTo change password, there will be three textboxes on the page, one for user to enter the password, two for user to enter the new password. For the change to succeed, the user must fulfil two requirements: (i) the current password user entered must match the one in the database and (ii) the password in the two new textboxes must match. Upon successful completion, the new password will be updated in the database. The error message if the new password user entered does not match the other textbox is shown in figure 33 below.

Figure 33 - Changing Password Error Message

# 

# Experiments and Results

This chapter will discuss the experiment and result for T5 model. The CNN/DM dataset method in chapter 4 will be used for this experiment. The purpose of this experiment is to make a comparison between the T5 (‘t5-base’) and the T5 (‘fine-tuned’) model for abstractive summarization.

## Dataset

Text

Description automatically generatedThis dataset is an English language dataset containing over 300k unique news articles as written by journalist at CNN and the Daily Mail. [19] The dataset consists of human-written summaries allowing it to be used for summarization tasks. Unlike the Gigaword and DUC dataset for abstractive summarization that only provides one sentence in each summary, CNN/DM dataset comprises of multi-sentence summaries. The source documents in the training set have 766 words spanning 29.74 sentences on average while the summaries consist of 53 words and 3.72 sentences. As show in figure 34, the CNN/DM consists of news articles and highlight sentences, Therefore, using the highlight as ground truth allows the T5 model to perform supervised learning.

Figure 34 - CNN/DM Raw Example (id: 42c027e4ff9730fbb3de84c1af0d2c506e41c3e4)

### Pre-process Data

Since T5 follows the text-to-text framework [16] and as shown in figure 3, to summarize a document for T5 model a prefix of ‘summarize’ needs to be added to the CNN/DM dataset before fine-tuning of model. Subsequently, T5 tokenizer will be applied to the news article resulting in a dictionary object containing an input ID and attention mask arrays containing the token id and attention masks for individual article.

As for the highlight or ground truth, T5 tokenizer will be applied to it and the same object dictionary will be like the input.

After pre-processing stage, the data can be used for fine-tuning and experiments that were discussed in the following chapter.

## Experimental Settings

Hugging Face [19] is an NLP focused Python library and it will be used in this experiment. The pre-trained model T5 (‘t5-base’) from Hugging Face will be utilized for abstractive summarization fine-tuning. The T5 (‘t5-base’) consists of roughly 220 million parameters. [16]

The input token length will be truncated to 512 tokens for T5 model because of the limitations of BERT that was used in HIWESTUM model. This is the learning rate used:

The model was trained with one epoch in remote Linux server consists of 4 GPU (GeForce GTX 3090) with each GPU having 24 GB allocated memory usage.

## ROUGE Evaluation Metrics

This experiment will be using Recall-Oriented Understudy for Gisting Evaluation (ROUGE) [19]. It is a metric that is commonly used to assess the quality of summarization and machine translation. The quality is determined by comparing the machine-generated summary and a human-generated summary. To make it easier to understand ROUGE, it is important to have prior knowledge of N-grams and metrics such as recall, precision and F1 scores.

N-grams are continuous sequences of words or tokens or neighboring sequences of items in a document where N denotes the length of the sequence. For example, N = 1 is a single word known as unigram and N = 2 is a sequence consisting of two consecutive words in order known as bigram.

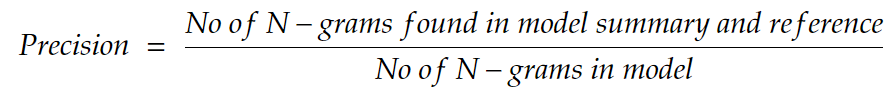
The standard method for calculating recall, precision and F1 scores is as follow:



In the formula, TP stands for True Positive, FP stands for False Positive and FN stands for False Negative. In the context of ROUGE, the method for calculating F1 scores remains the same but the computation of recall and precision is as follow:

A picture containing letter

Description automatically generated



Using simplified notation for recall and precision:



Therefore, ROUGE-N can be computed using the formula below:

Schematic

Description automatically generated with medium confidence

For instance, if we need to calculate the ROUGE-2 score for the following summaries:

|  |
| --- |
| Summary by Model:  I love data science |
| Gold Summary:  I really love data science |

The bigram tokens extracted from the model summary and gold summary would be [‘I love’, ‘love data’, ‘data science’] and [‘I really’, ‘really love’, ‘love data’, ‘data science’] respectively.

The metrics can be computed as follow:





## T5 Experiments

This section will be comparing the performance of T5 base and fine-tuned T5 model for document summarization by using the ROUGE metric mentioned in Section 5.2 and gold summary (ground truth). The experimental settings hyperparameters mentioned in Section 5.1 and the CNN/DM dataset mentioned in Chapter 4 Section 2 will be used for training, validation and testing for the finetuning of T5 model for summarization. Table 4 below shows the ROUGE scores of the fine-tuned T5 model with only one epoch.

|  |  |  |  |
| --- | --- | --- | --- |
| Fine-tuned Model | ROUGE-1  (R1) | ROUGE-2  (R2) | ROUGE-LSUM  (RL) |
| T5 (CNN/DM) | 28.3647 | 13.9911 | 26.6136 |

Table 4 - T5 Rough Score

To compare the differences between T5 (‘t5-base’) and T5 (‘fine-tuned’), a test data will be extracted from the CNN/DM and comparison between the two outputs from the model and the gold summary is shown in the table below.

|  |
| --- |
| Summary by T5 (‘t5-base’):  the formal accession was marked by a ceremony at the Hague, where the court is based. the ICC opened a preliminary examination into the situation in Palestinian territories. |
| Summary by T5 (‘Fine-Tune’):  The Palestinian Authority officially becomes the 123rd member of the International Criminal Court. The court has jurisdiction over alleged crimes in Palestinian territories. Israel and the United States opposed the Palestinian |
| Gold Summary:  Membership gives the ICC jurisdiction over alleged crimes committed in Palestinian territories since last June. Israel and the United States opposed the move, which could open the door to war crimes investigations against Israelis. |

# 

# Conclusion and Future Work

## Conclusion

The project intended to create a web application for document summarization to provide users with a better understanding of document summarization and a portal for them to experience the process of summarizing a document. The web application is user-friendly and the simple user interface is designed to be easy to use and navigate with the goal of helping users accomplish their tasks quickly. The Python Flask web framework was chosen for its simplicity and flexibility. Additionally, using Python can allow swift and straightforward integration with the HIWESTSUM model.

The final product allows users to access and learn about the two techniques of document summarization and understand more about the architecture this project used without the need to create an account. Additionally, users can utilize the summarizer by uploading a document in either PDF or TXT format and selecting the cumulated probabilities allows the user to select the number of sentences to return from the extractive summarizer. Subsequently, the output from the extractive summarizer will be used as input for the abstractive summarizer to process and produce the final result.

The original document, top K sentences from the HIWESTSUM model and final result from the T5 model will be stored in a database. By doing so, users can use the previous summarization result to compare with another method of summarization or use it as a reference for their own usage.

## Recommendation in Future Work

### Web application recommendation

Front-end Framework

With the recent advancement of frameworks or libraries like React.js, which is an open-source JavaScript framework developed by Facebook. React.js is usually utilized for building interactive interfaces and web applications quickly and efficiently with significantly less coding. Basically, React.js enables the development of large and complex web-based applications that do not refresh the page when data is change.

Utilizing reusable UI components in React.js allows usage across different pages of the web application, resulting in improved speed and efficiency during coding and being less prone to errors. Additionally, React.js does not refresh the current web page when data changes. This can be utilized by implementing a function that allows users to pick the highly scored sentences that they prefer instead of using cumulated probabilities to return the top K sentences.

Furthermore, API can be utilized because it is platform independent, allowing integration with other systems in the near future. APIs support caching, security, and multiple data formats, which makes them a popular choice for modern web application development.

Back-end Framework

Instead of using a traditional approach to web programming, the implementation of cloud computing can be considered because it can improve scalability, relatability and performance. A recommendation would be to use cloud database services like Amazon Web Services (AWS) RDS, Google Cloud Platform (GCP) Cloud SQL and Azure SQL. Cloud platforms provide services like automatic backup, failover and scaling for varying levels of traffic.

Additionally, cloud-based monitoring and logging services like AWS CloudWatch, GCP Stackdriver and Azure Monitor can be utilized to monitor web application metrics, logs and other important data. This allows visibility of the performance and health of the web application.

Overall, by leveraging cloud services, web application performance, scalability, reliability and security can drastically improve but it is important to choose the right cloud services to fit the web applications needs.

### Document Summarizer

With the recent advances of CHATGPT [1], GPT-3[2] and the upcoming GPT-4, there is a need to understand and research on generative artificial intelligence and prompt engineering as well. The reason is that, because of the limitations of extractive summarization and the HIWESTSUM model is trained to maximize ROUGE score between the result and gold summary. On the other hand, CHATGPT or GPT-3.5 and GPT-3 are pre-trained on massive amounts of text data using unsupervised learning techniques and learning the structures of language then allowing the model to be fine-tuned for downstream tasks through prompt engineering.

Additionally, GPT models can overcome the challenges of encoding longer documents that far exceed the BERT limitations of 512 tokens.

Finally, the GPT model can be used to perform other downstream tasks that can be utilized in the web application and adapted into a general NLP web application rather than focusing on document summarization.

## Learning Reflection

This Final Year Project was my first project consisting of web development and natural language processing. It was a wonderful and enriching experience for me, and I appreciate the process. This experience has further enhanced my development skills by teaching me how to develop a full-stack web application and train a machine learning model that I have never tried before. Additionally, several approaches and methodologies were adopted towards the development of this web application. I hope that through this web application, users will be able to learn about document summarizers and at the same time experience the process of summarization.

One of the things I learned is that choosing the right technological stack is important, but it can be a daunting task because of the numerous options available. Each stack comes with its strengths and weaknesses. To help me make an informed decision, it is essential for me to consider some factors such as the requirements, time and available resources. This freedom of choice allows me to leverage cutting edge technologies and make this project realizable.

This project has imparted learning opportunities for me on various tools and technologies. I have gained knowledge on working with Python Flask framework, PostgreSQL and fine-tuning an NLP model for downstream tasks. Additionally, I gained proficiency in using Linux commands and sessions to execute my training.

Throughout the design and development of this project, I have matured into a self-directed and self-motivated learner and have become more dynamic and adaptive when it comes to problem solving. Software development, natural language processing and product management knowledge are the major takeaways for my Final Year Project. As an undergraduate at NTU, I am grateful for this opportunity.

Finally, I would like to thank my project supervisor, A/P Chen Li Hui and PhD in-charge Xu Jia Hao for their guidance and providing invaluable advice throughout the project. This project has given me the opportunity to learn, and I have learned a lot in this process.

# References

[1] M. Aljanabi, M. Ghazi, A. H. Ali, and S. A. Abed, “*View of CHATGPT: Open possibilities.*”, 2023

[2] A. Radford, K. Narasimhan, T. Salimans and I. Sutskever, "I*mproving Language Understanding by Generative Pre-Training*", 2018.

[3] Y. Liu, "*Fine-tune BERT for Extractive Summarization*", 2019

[4] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu, “*Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer*” 2020.

[5] J.-ge Yao, X. Wan, and J. Xiao, “*Recent advances in document summarization - knowledge and information systems*,” SpringerLink, 28-Mar-2017

[6] T, Rui Jie, “*Hierarchical Document Representation for Summarization*”, EE4080 Final Year Project report. Unpublished manuscript, School of Electronics and Electrical Engineering, Nanyang Technological University, pp, 18-37, 2022

[7] Krause Jörg, *Introducing web development*. Berkeley, pp, 39, 2016

[8] C. Buckler, *Modern CSS*. Collingwood, Australia: SitePoint Pty, Limited, pp, 1, 2018

[9] S. Sumathi and S. Esakkirajan, *Fundamentals of Relational Database Management Systems*. Berlin: Springer, 2007.

[10] N. Jatana, S. Puri, M. Ahuja, I. Kathuria, and D. Gosain, “*A Survey and Comparison of Relational and Non-Relational Database*”, 2012

[11] Mohd, M., Jan, R., & Shah, M. (2020). *Text document summarization using word embedding*. *Expert Systems with Applications*, *143*, 112958. https://doi.org/10.1016/j.eswa.2019.112958

[12] M. Gambhir and V. Gupta, *Recent automatic text summarization techniques: a survey*. 2016.

[13] S. Gupta and S. K. Gupta, “Abstractive summarization: *An overview of the state of the art*,” 07-Dec-2018.

[14] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, “*Attention is all you need*,” *arXiv.org*, 06-Dec-2017.

[15] A. Malte and P. Ratadiya, “*Evolution of transfer learning in natural language processing,*” *arXiv.org*, 16-Oct-2019.

[16] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu, “*Exploring the limits of transfer learning with a unified text-to-text transformer*,” *arXiv.org*, 28-Jul-2020.

[17] R. Fielding, U. C. Irvine, and J. Gettys, “*Hypertext Transfer Protocol -- HTTP/1.1,*” 1999.

[18] K. Douglas and S. Douglas, *PostgreSQL: The Comprehensive Guide to building, programming, and administering postgressql databases*. Indianapolis, IN: Sams Publishing, 2005.

[19] C. Lin, "*ROUGE: A Package for Automatic Evaluation of Summaries*", 2004

# Appendix (optional)