Informatics Institute of Technology

In Collaboration With

University of Westminster, UK



*University of Westminster, Coat of Arms*

GATOT

Design

Mr. Nazhim Kalam

w1761265 / 2019281

Supervised by

Mr. Torin Wirasingha

September 2022

This Project Proposal is submitted in partial fulfilment of the requirements for

the BSc (Hons) Computer Science degree at

the University of Westminster.

Table of Contents

[List of Figures ii](#_Toc125462559)

[List of Tables ii](#_Toc125462560)

[1. Chapter Overview I](#_Toc125462561)

[2. Technology Selection I](#_Toc125462562)

[2.1 Technology Stack I](#_Toc125462563)

[2.2 Data Selection I](#_Toc125462564)

[2.3 Selection of development framework I](#_Toc125462565)

[2.4 Programming Language I](#_Toc125462566)

[2.5 Libraries Utilized I](#_Toc125462567)

[2.6 IDE’s Utilized I](#_Toc125462568)

[2.7 Summary of Technology selection I](#_Toc125462569)

[3. Implementation of Core Functionalities I](#_Toc125462570)

[4. Testing & Evaluation Code of Models II](#_Toc125462571)

[5. User Interface II](#_Toc125462572)

[6. Chapter Summary II](#_Toc125462573)

[References II](#_Toc125462574)

# List of Figures

[Figure 12.1 - Prototype Feature Diagram (Self-composed) 13](#_Toc117550682)

[Figure 13.1 - Gantt Chart 16](#_Toc117550683)

[Figure 13.2 - Model development flow (Self-composed) 21](#_Toc117550684)

# List of Tables

[Table 5.1 - Related work in abstractive text summarization 3](#_Toc117584436)

[Table 11.1 - Research Objectives 9](#_Toc117584437)

[Table 13.1 - Research Methodology 13](#_Toc117584438)

[Table 13.2 - Deliverables and Dates 17](#_Toc117584439)

[Table 13.3 - Risk Mitigation Plan 19](#_Toc117584440)

**Acronyms**

|  |  |
| --- | --- |
| AI | Artificial Intelligence. |
| DL | Deep Learning |
| GUI | Graphical User Interface |
| ML | Machine Learning |
| NLP | Natural Language Processing |
| ROUGE | Recall-Oriented Understudy for Gisting Evaluation. |
| BLEU | BiLingual Evaluation Understudy. |
| T5 | Text to Transfer Transformer. |
| BART | Bidirectional Auto-Regressive Transformers. |
| BERT | Bidirectional Encoder Representations from Transformers. |
| PEGASUS | Pre-training with Extracted Gap-sentences for Abstractive Summarization Sequence-to-sequence |
| ILP | Inductive logic programming. |
| LSTM | Long Short-Term Memory. |
| RNN | Recurrent Neural Network. |
| CNN  SEQ2SEQ | Convolutional Neural Network.  Sequence to Sequence |
| RoBERTa | Robustly Optimized BERT Pre-training Approach |
| GPT-3  REST  GPU | Third Generation Generative Pre-Trained Transformer  Representational State Transfer  Graphical Processing Unit |

# Chapter Overview

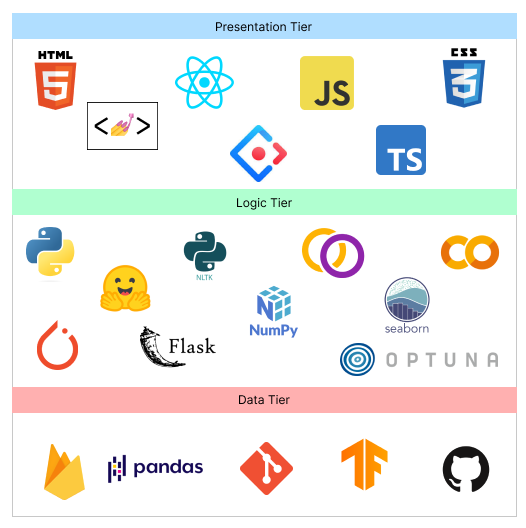
This chapter will provide a thorough overview of the technologies, supporting tools, and languages utilized for the project development, as well as the fundamental implementation of the research prototype.

# Technology Selection

## **Technology Stack**

The technologies utilized to implement the prototype at each tier are given below.

Figure 2.1: Technology Stack (*self-composed*)



In preference to macOS and Linux, **Windows** will be the operating system used for project development and documentation. This is due to a wider variety of software available, which ensures that it has more industry-standard tools than Linux and macOS, along with better compatibility and familiarity, which make things simpler to use and manage.

## **Data Selection**

Given that the project relies heavily on data science, it is essential to use data from trustworthy sources to train the model. This ensures that the data is accurate and leads to the development of a more accurate model for general text summarization.

The goal of the project was to develop an adaptive generalized text summarization model, so a generalized dataset for text summarization was necessary to establish the base model. **TensorFlow datasets**, being a reputable source of data, offered multiple options for this dataset.

Previous research has suggested that the **CNN Dailymail** dataset, **Gigaword** dataset, and **Xsum** dataset are well-suited for general text summarization. These datasets are available on TensorFlow datasets and were considered as a good choice for the project.

During the training process, all three of these datasets (CNN Dailymail, Gigaword, and Xsum) were utilized with various transformer architectures to determine which dataset resulted in the best evaluation metrics. Of the three datasets, Xsum performed the best, so it was selected as the final dataset for the project.

## **Selection of development framework**

Table 7.1: Development framework utilized

|  |  |
| --- | --- |
| **Framework** | **Reason for choosing** |
| React | ReactJS provides reusable components for efficient application development, and its open-source nature and strong community support enable continuous developments and learning tools, making it a handy solution for developers. |
| Ant Design | Ant Design is a popular React UI framework that offers a large selection of pre-built components, encourages consistency and usability, and enables for style customization using CSS-in-JS. It also reduces build time by using tree-shaking compatibility. Overall, it provides a complete and effective frontend development solution. |
| Flask | Flask is a Python micro web framework that is lightweight, easy to learn, and provides for flexibility in developing application structures. It is useful for developing backend APIs since it provides a straightforward approach to manage routing and request processing, as well as a built-in development server and different extensions that can be used to extend an API's capabilities. |
| Optuna | Optuna is a Python open-source framework for hyperparameter optimization that is simple to use, efficient, and has built-in parallelization support. It also offers built-in support for popular machine learning libraries, as well as automated early halting and distributed parallel optimization. It is a robust and adaptable library that can aid in the improvement of machine learning model performance. |
| PyTorch | PyTorch is a Python open-source machine learning framework that is built on Torch and makes use of GPU capability. Because of its straightforward and easy-to-use API, vast selection of pre-built neural network layers and modules, powerful features such as dynamic computation graphs and automated differentiation, and strong community support, it's a solid choice for developing machine learning models. It is widely used in business and academia for machine learning model research and development. |

The data science core employs transformer models from Hugging Face, which have been fine-tuned with the datasets used in this research project. The purpose of retraining the model is to experiment with various hyperparameter changes.

## **Programming Language**

In this study, we employed the programming language **Python** for the implementation of our Machine Learning models and Backend APIs. Python is a widely-used language known for its readability, simplicity and versatility, making it an ideal choice for our research project. This language has a broad range of use cases including web development, data analysis, scientific computing and machine learning. Additionally, Python has a large and active community, providing ample resources and support. Furthermore, the availability of various libraries and frameworks such as NumPy, pandas, and TensorFlow, made Python a powerful tool for our data science and machine learning tasks.

**JavaScript** was chosen for the frontend development in order to display dynamic content and create a highly interactive and engaging user experience.

## **2.5 Libraries Utilized**

Table 7.2: Libraries used with reasonings

|  |  |
| --- | --- |
| **Library** | **Reasoning for selection** |
| React | Used to build user interfaces for web applications using a declarative approach. |
| Firebase | Used for providing backend services for mobile and web application development. |
| TypeScript | Used to enhance JavaScript and make it appropriate for building large-scale applications by adding optional static typing and other features like class and interface. |
| Axios | Used for handling HTTP requests in JavaScript. |
| Redux | Used to control the state of JavaScript applications in a predictable manner by the use of actions, reducers, and a central store. |
| Optuna | Used for performing hyperparameter optimization, and it is used to find the best set of hyperparameters for machine learning models. |
| Torch | a library for machine learning and deep learning that provides a powerful set of tools for building and training neural networks, particularly in computer vision and natural language processing tasks. |
| Hugging face Transformers | Hugging Face transformers library is a state-of-the-art natural language processing library that provides pre-trained transformer models and tools for fine-tuning them on specific tasks. |
| NLTK | NLTK is a library for natural language processing that provides tools for tasks such as tokenization, stemming, and part-of-speech tagging, as well as a wide range of corpora and resources for training and evaluating language models. |
| Rouge | A library for evaluating the quality of text summaries, it is used to compare an automatically generated summary or a peer summary to one or multiple reference summaries. |
| Pandas | Pandas is a library for data manipulation and analysis, it provides data structures and data analysis tools for handling and manipulating numerical tables and time series data, it is widely used for data preprocessing and data cleaning tasks in data science. |
| NumPy | NumPy is a library for scientific computing with Python, it provides support for large, multi-dimensional arrays and matrices of numerical data, as well as a large collection of mathematical functions to operate on these arrays |
| Matplotlib & Seaborn | Used for creating static, animated, and interactive visualizations in Python |
| Gramformer | Used for generating text using GPT-3 model, it's developed by Hugging Face. It provides an easy to use API that allows developers to fine-tune GPT-3 models on their own data and use them for text generation, it supports for various tasks such as text completion, text generation, and text classification. |
| Flask | Used for creating web APIs using Python to communicate with the transformer model and handling HTTP requests. |

## **2.6 IDE’s Utilized**

Table 7.3: IDEs Utilized with justification for choices

|  |  |
| --- | --- |
| **IDE** | **Justification for selection** |
| VSCode | Best known for its adaptability, usefulness, and performance, it offers a wide range of capabilities, such as debugging, Git integration, syntax highlighting, and extensions to personalize the environment. |
| Google Colab | Due to its connection with Google Drive and availability of free GPUs, it’s helpful for developing machine learning models via a cloud environment. |
| Jupyter Notebook | Due to their interactive and readable format, making it ideal for local experimentation, documentation and collaboration. |

## **2.7 Summary of Technology selection**

Table 7.4: Summary of Technology selection

|  |  |
| --- | --- |
| **Component** | **Tools** |
| Programming Languages | JavaScript, Python |
| Development Framework | Flask, PyTorch |
| UI Framework | Ant Design, React |
| Libraries | NLTK, Rouge, React, Pandas, Optuna |
| IDE – Research | Google Colab, Jupyter Notebook |
| IDE – Product | VSCode |
| Version Control | Git, Github |
| Application hosting | Firebase, Render |

# Implementation of Core Functionalities

Once code is completed will get into this

# Testing & Evaluation Code of Models

Once code is completed will get into this

# User Interface

The UI wireframes related to the MVP (Minimum Viable Product) has been listed under the **Appendix D**

# Chapter Summary

The chapter discusses the tools, technology, and languages utilized to create the research prototype. The fundamental functionality is covered, along with insights and samples of code for the implemented algorithms. Also highlighted are the planned user interfaces for the minimal viable product project.

# References