Informatics Institute of Technology

In Collaboration With

University of Westminster, UK



*University of Westminster, Coat of Arms*

Generalized Abstractive Text Summarization Using Optimized Transformers

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](#_Toc120522959)

[List of Tables ii](#_Toc120522960)

[1. CHAPTER OVERVIEW 1](#_Toc120522961)

[2. DESIGN GOAL 1](#_Toc120522962)

[3. HIGH-LEVEL DESIGN 1](#_Toc120522963)

[4. SYSTEM DESIGN 1](#_Toc120522964)

[4.1 Choice of the Design Paradigm. 1](#_Toc120522965)

[4.2 Data Flow Diagram 1](#_Toc120522966)

[4.3 Algorithm Design 1](#_Toc120522967)

[4.4 UI Design 1](#_Toc120522968)

[5. CHAPTER SUMMARY 2](#_Toc120522969)

[REFERENCES I](#_Toc120522970)

# 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

The design choices taken to create a suitable architecture for implementation, depending on the requirements received, are discussed in this chapter. To explain how the design goals are intended to be accomplished while outlining the justification for selected design decisions, high-level design, low-level design, design diagrams, and UI wireframes have been utilized.

# Design Goals

Table 6.1 – Design Goals of the proposed system

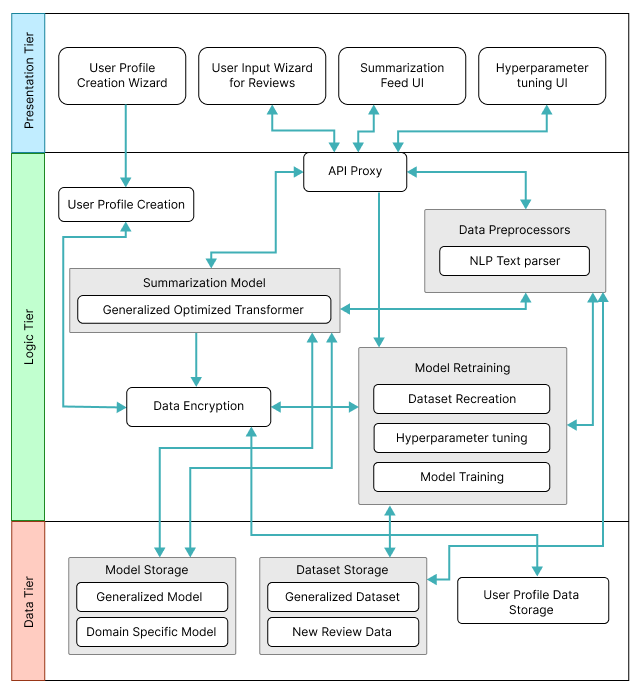
|  |  |
| --- | --- |
| **Design Goal** | **Description** |
| Performance | To find the new set of hyperparameters with the new data, model retraining requires a significant amount of time. As a result, the newly created dataset (with unseen data) should be accurately made, and it is best if it takes the least amount of time to query the data from various businesses within the same domain to create the dataset. Moreover, other core functionalities should be designed effectively to increase overall performance. |
| Correctness | The correctness & quality of the output should be of the highest possible level utilizing the optimized transformer architecture. Since several approaches are considered in order to get the optimized solution the expected output should of the best possible form. |
| Usability | The system's usability must be straightforward for users of all levels of knowledge because its primary function is to summarize review text for any domain, including movies and general users. |
| Scalability | In a production environment, the system may need to accommodate a large number of concurrent user requests. This should be manageable by the backend. The system should be easily expandable to accommodate new data. |
| Adaptability | Adopting new features or components need to be a simple procedure. The system shouldn't be broken if a component is added or removed, and it shouldn't be affected overall. |

# High-level Design

## **3.1 Tiered Architecture**

The image below depicts the architecture of the system. Three tiers of architecture separate the data, logic, and presentation levels. The system's generalization and domain specific adaptive hyperparameter tuning and data preprocessing represent the research contribution.

Figure 3.1: Three-Tiered Architecture (*self-composed*)



# System Design

## **Choice of the Design Paradigm.**

## **Data Flow Diagram**

In order to show the relationships between components and provide a clearer understanding of how data flows across the whole system, the context diagram's components have been extensively broken down in the diagram below, which was detailed in the SRS previously.

## **Algorithm Design**

## **4.4 UI Design**

# Chapter Summary

# References