Informatics Institute of Technology

In Collaboration With

The University of Westminster, UK

The University of Westminster, Coat of Arms

GenSum

Adaptive Generalized Text Summarization System using Optimized Transformers

A dissertation by

Mr. Nazhim Kalam

W1761265 | 2019281

Supervised by

Mr. Torin Wirasingha

April 2023

Submitted in partial fulfilment of the requirements for the

BSc (Hons) Computer Science degree at the University of Westminster.

DECLARATION

I affirm that this dissertation, including its sub-components, is the product of my own original research endeavors. Furthermore, I confirm that I have not previously submitted or presented any of this content, in whole or in part, as part of any other degree or qualification program at any other university or institution. Any factual information obtained from credible external sources has been duly acknowledged through proper citation.

Student Name: Nazhim Kalam

Registration Number: w1761265 -- 2019281

Signature: **Date:** April 27, 2023

ABSTRACT

Abstractive text summarization systems have been integrated with various application in the world to perform text summarization and its nothing new to the field. However, with the prior research it found that in the domain of movies the need for performance improvement is required using latest approaches than the current traditional ML & DL methods, movie review summarization plays a major role in helping users to make better decisions by matching their interest with the reviews of the movie, this saves a lot of time and also improves businesses in their sales.

In 2017 researches from Google Brain introduced NLP Transformers, which is a latest approach to solve NLP problems and its increasingly been known and used nowadays over traditional ML & DL approaches like using basic LSTM, RNN approaches. The author explored ways in which to get an optimal solution using Transformer for abstractive text summarization and yet making a generalized solution which can be adapted with respect to any domain (be it hotels, movies, restaurants) and increase its performance as the system gets used over with time.

The author was able to experiment with few of the top tier transformer architectures to filter out the optimal model and integrated an automated hyperparameter searching mechanism which will find the best set of hyperparameters to train the model with respect to any domain. **ROUGE1 of 80.8, ROUGE2 of 79.42, ROUGEL of 80.8, ROUGELSUM of 80.8** was the optimal evaluation metric result achieved from the BART model giving the best result.

Keywords: Natural Language Processing (NLP), Machine Learning (ML), Deep Learning (DL), Recall-Oriented Understudy for Gisting Evaluation (ROUGE), Inductive logic programming (ILP)

Subject Descriptors:

- Computing methodologies → Artificial intelligence → Natural language processing → Natural language generation
- Theory of computation → Theory and algorithms for application domains → Machine learning theory → Semi-supervised learning.
- Information systems → Information systems applications → Management and querying of encrypted data.
- Security and privacy → Database and storage security → Data mining.

PUBLICATIONS

1. **An Adaptive Approach for Generalized Text Summarization Using Optimized Transformers**

   Conference: SmartNets 2023 – Big Data Analytics and Machine Learning

   Status: Submitted, In-Review

   Venue: Asia, Colombo

   Date: April 25, 2023

2. **A Review on Creating a Performance Adaptive Generalized Abstractive Text Summarization Using Optimized Transformers**

   Conference: SmartNets 2023 - Big Data Analytics and Machine Learning

   Status: Submitted, In-Review

   Venue: Asia, Colombo

   Date: April 08, 2023

APPENDIX L shows the proof of Research & Review paper submission made at SmartNets 2023 Conference.

ACKNOWLEDGEMENT

I am grateful for the supportive individuals in my life who have empowered me to exceed my previous limitations and achieve my current level of success, including the completion-of this final year research project.

I would like to express my deepest gratitude to Mr. Guhanathan Poravi, my esteemed module leader, for his consistent review and invaluable guidance that has tremendously improved my reports. His feedback has instilled in me the confidence to conduct research at any level. I am also grateful to my supervisor, Mr. Torin Wirasingha, for his contribution in refining my project idea and expanding its scope. I extend my appreciation to the evaluators of this project, who took the time to thoroughly review my work and provide valuable advice for its enhancement. Furthermore, I would like to thank all the interviewees and survey respondents, including industry and academia experts, as well as my senior and junior peers at IIT, for their input in the requirement gathering process.

I express my heartfelt appreciation to the esteemed lecturers at the Informatics Institute of Technology, as well as the supportive management, for the opportunities, unwavering support, and guidance they have provided me throughout my undergraduate degree. I would also like to acknowledge my fellow friends from university, whose continued support and positive influence have played a pivotal role in my growth as a developer/researcher. Their encouragement and motivation have fueled my progress over the span of four years. Lastly, I extend my deepest gratitude to my parents for their support and encouragement in pursuing my dreams, and to my beloved family, friends, and significant other, whose encouragement has been a strength during challenging times.

CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

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	Convolutional Neural Network.
SEQ2SEQ	Sequence to Sequence
RoBERTa	Robustly Optimized BERT Pre-training Approach
GPT-3	Third Generation Generative Pre-Trained Transformer
REST	Representational State Transfer
GPU	Graphical Processing Unit
API	Application Programming Interface

CHAPTER 01. INTRODUCTION

1.1 Chapter Overview

In this chapter, a series of top-tier pretrained transformer designs are optimized using automated search hyperparameter optimization in an effort to improve the performance of abstractive text summarization for movie reviews while developing a generalized solution that can adapt and be used in other domains. Along with a review of previous studies and a presentation of the anticipated project timetable, the research problem, gap, challenge, and method will be discussed in the work plan.

1.2 Problem Domain

1.2.1 Movie User Reviews

A growing number of websites, like Amazon and the Internet Movie Database (IMBD), a website for movie reviews, allow users to publish reviews for things they are interested in, along with the growth of Web 2.0, where user interaction is prioritized. (Khan, Gul, Zareei, et al., 2020)

Online movie reviews are evolving into an important information source for users, with the continuous increase in data on the web (M and Mehla, 2019). However, online users post a significant number of movies reviews every day, hence making it difficult for them to manually summarize the reviews and determine their interest in the film. One of the challenging problems in natural language processing is mining and summarizing movie reviews. (Khan, Gul, Uddin, et al., 2020).

Text summary assist users or business decision-makers by compiling and analyzing a significant number of online reviews. (Alsaqer and Sasi, 2017). These days, the majority of people research a film's reviews before selecting or watching it on any platform, such Netflix or Amazon Prime, but we also come across conflicting reviews that can be either good or bad. While most reviews are detailed and require a significant amount of time to review, this develops a problem where users aren't able to make quicker decisions. Therefore, by summarizing the review makes it easier and faster for users to make decisions. This can also help streaming services like Netflix quickly discover the viewing habits or preferences of their users (Dashtipour et al., 2021)

1.2.2 Text Summarization

Today, there is a lot of textual material available, including news stories and reviews. Text summarizing helps us quickly find the key elements of the full piece by minimizing the quantity of text. (Mahajan et al., 2021).

Extractive summarization and abstractive summarization are typically the two methods of text summarization. When extractive summarizing, the most important lines from the context or article are plucked out without being altered in any way. Meanwhile, abstractive summarizing aims to create the sentences on its own and creates the summary; this is superior than extractive summarization since it is more meaningful to generate our own phrases inside the context rather than to utilize selected sentences from the context without any change. (Etemad, Abidi and Chhabra, 2021).

1.2.3 Transformers

Transformers in NLP is a novel architecture that aims to solve sequence-to-sequence tasks while handling long range dependencies with ease. It has surpassed competing neural models like CNN (Convolutional Neural Nets) and RNN (Recurrent Neural Nets) in terms of performance to appear as the dominant architecture for natural language processing (Wolf et al., 2020).

Transformers uses self-attention mechanism to target on selected areas of the input sentence followed by the encoder and decoder architecture (Etemad, Abidi and Chhabra, 2021).

1.3 Problem Definition

In the domain of movie review summarization, currently there are no researches done using the latest deep learning approaches (***such as Transformers***) to solve this problem, standard machine & deep learning algorithms such as Naïve Bayes, RNN have been used, the usage of advanced deep learning approaches can be utilized in order to enhance the quality/accuracy of the text summarization.

Deep learning models take longer to train but they provide greater accuracy since they can simultaneously automate feature extraction and classification, whereas machine learning algorithms require feature selection at first. Therefore, applying deep learning techniques will help to improve the quality of text summarization and help the user in making better decisions (Etemad, Abidi and Chhabra, 2021).

1.3.1 Problem statement

No prior research has looked into applying cutting-edge deep learning methods like Transformers to produce abstractive summaries from movie reviews, which can improve text summarization. This solution aims to be generic and accessible to any sector. (Khan, Gul, Zareei, et al., 2020).

1.4 Research Motivation

The identified problem can also be applied to several other domains which requires to improve the quality abstractive text summarization using the advanced approaches of deep learning, not only specific movie reviews, this is why a generalized solution was thought of initially (Kouris, Alexandridis and Stafylopatis, 2019).

As mentioned in the work of (Etemad, Abidi and Chhabra, 2021), syntactic and semantic issues with text summarization were the main issues that researchers were concerned on solving. and with respect to their research by exploring multiple deep learning techniques, they concluded that Transformer based models (T5 model) outperformed in all NLP tasks, this encourages the author to go deeper into the field of transformers optimization in order to enhance the quality of text summarization and address the constraints associated with the summarizing of movie reviews.

1.5 Research Questions

RQ1: What are the top tier transformer architectures widely used and know for NLP problems related to text summarization?

RQ2: How can a pretrained transformer architecture be fine-tuned to get the optimal hyper parameters and to automate it for model retraining?

RQ3: What kind of evaluations should we perform after fine-tuning to filter out the best transformer architecture?

RQ4: How can domain generalization be integrated for system?

1.6 Research Aim & Objectives

1.6.1 Research Aim

The aim of this research is to design, develop and evaluate an optimal adaptive generalized transformer architecture from a range of popularly used architectures by fine-tuning via automated hyperparameter optimization, therefore obtaining the recommended architecture's optimum performance

A fully working system that can be utilized to perform abstractive text summarization based on the user input from any domain (movie, hotel, ecommerce etc.…) will be created by this research project. The quality of the resulting text summary or performance optimization will be the **main points** of emphasis. To get the best result, the usage of data preparation, data analysis, conducting hyperparameter tuning, and evaluating the models will be investigated.

Components will be built, necessary information will be gathered and researched, and performance will be assessed. The system may be utilized for private or public purposes on both a hosted server and a local browser. The data science models' source code will be made available in a public repository for future research and use. A research paper will be published and it end of this study.

1.6.2 Research Objectives

For the research to be considered successful, its goals must be fulfilled

Table 1: Research Objectives (Self-Composed)

Objective	Description	LO	RQ
Problem Identification	Comprehend and document the identified issue. **RO1:** Perform research in a domain of interest and identify a sufficiently comprehensive problem that needs to be addressed. **RO2**: Thoroughly explore and analyze potential solutions for addressing the problem.	LO1, LO2	RQ1, RQ4

	RO3: Explore methods for developing an adaptive and generalized approach. **RO4**: Create a schedule, determine associated deliverables, and develop a Gantt chart for the project.		
Literature Review	Complete a thorough critical review of earlier related work. **RO1:** Make a preliminary investigation on existing abstractive text summarization using deep learning approaches. **RO2:** Make a preliminary investigation on why transformers architecture was the chosen deep learning choice for this research. **RO3:** Analyze the top tier transformer architectures widely used. **RO4:** Analyzing how the models can be fine-tuned via hyperparameter optimization. **RO5:** Analyzing the different approaches used for model evaluation. **RO6:** Analyze how the model can be generalized for every other domain.	LO1, LO4, LO8	**R**Q1, **R**Q2, **R**Q3, **R**Q4
Methodology Selection and SLEP Framework	This defines the outline structure for the requirement analysis and the design process followed by the social legal ethical and professional issues. **RO1**: Analyzing the Research Methodology approaches. **RO2**: Analyzing the Development Methodology approaches. **RO3**: Analyzing the Project Management Methodology approaches.	LO2, LO6	**R**Q4, **R**Q2, **R**Q1

	RO4: Analyzing the Solution Methodology approaches. **RO5**: Analyzing the Social, Legal Ethical and Professional Issues which could develop during the phase of the project.		
Requirement Elicitation	Defining the project's needs utilizing relevant approaches and tools in order to solve the projected research gaps and obstacles based on prior related research. **RO1:** Gathering information related to the expected metadata required for the dataset to contain for the model training. **RO2:** Gathering the requirements of transformer architectures for fine-tuning and understand the end to end user expectations. **RO3:** Getting insights from domain experts to build a suitable system. **RO4:** Gathering the requirements for handling generalization.	LO1, LO3, LO5	**R**Q4, **R**Q2, **R**Q1
Design	Considering the following when developing the suggested system: **RO1:** Design a component to preprocess the dataset for the respective model inputs. **RO2:** Design a component to store the top tier transformer models with their respective metadata, to use throughout. **RO3:** Design a hyperparameter tuning component that can improve accuracy of the transformer model. **RO4**: Design high-level architecture for the system.	LO1, LO5	**R**Q2

Implementation	Setting up a mechanism capable of addressing the gaps that were intended to be covered. **RO1**: To develop data preprocessing component. **RO2**: To develop a component that handles and stores the top tier transformer architectures for fine-tuning. **RO3**: To develop the automated hyperparameter search component that handles all the top tier architectures assigned. **RO4**: To develop a component for the model evaluations for the measured hyperparameters	LO1, LO5, LO7	**R**Q2, **R**Q3
Evaluation	Testing and evaluating the developed system (including the data science models with the suitable metrices) **RO1**: Performing unit test, integration and performance testing along with a test plan created. **RO2**: Evaluating all the transformer architectures used for fine-tune experimentations, using recommended scores such as (ROUGE or BLEU SCORE).	LO1, LO5	**R**Q3
Documentation	Keeping track of and documenting the study project's ongoing progress and any challenges encountered.	LO6, LO8	-
Publication	Ensure that the documentation, reports, and papers are well-structured and include a critical analysis of the research. **RO1**: To publish a research paper on the related work done. **RO2**: To publish the testing & evaluation results of the work done. **RO3**: To publish the code implementation repository as public to be access by future research investigations, along with the models and datasets	LO4, LO8	-

1.7 Novelty of the Research

1.7.1 Problem Novelty

The problem novelty of this research is, the lack of attempt to increase transformer performance in order to get better textual summarizing outcomes (Khan, Gul, Zareei, et al., 2020).

1.7.2 Solution Novelty

The solution novelty for this problem is performing an automated approach for hyperparameter tuning & creating a retraining mechanism with newly exposed data to enhance its performance further using the optimal transformer with respect to any domain (generalization) (Etemad, Abidi and Chhabra, 2021).

1.8 Research Gap

Based on previous work done (Khan, Gul, Zareei, et al., 2020) related to abstractive text summarization on movie reviews, the literature identifies for the need of using advanced deep learning approaches to improve the performance of text summarization for this movie domain over traditional machine learning approach.

This project focuses on Empirical gap in the Movie Domain, as well as Theoretical and Performance gaps in the area of transformer optimization. Transformers plays a major role in the field of deep learning especially at problems related to Natural Language Processing, by performing hyperparameter optimization on several transformer architectures we can contribute to the enhanced quality of abstractive text summarization and create a generalized model which can be adapted with the respective domains usage and improve the performance

1.9 Contribution to the body of knowledge

Improving the performance of an existing solution is very common in the field of data science, as we can explore new algorithms or fine-tuning existing algorithms to get better results. The contributions for this project can be classified as theoretical contributions and domain contributions.

The following is a summarization of the authors contribution:

- ***Abstractive Text Summarization:*** *Automated Hyperparameter optimization + Model Retraining + Transformers + Deep Learning*
- ***Movie User Review & Generalization:*** *Research domain target is for Movie reviews, in addition the author makes the system generalized to adapt to any domain area*.

1.9.1 Research Domain Contribution

There are various deep learning techniques that can be used to handle abstractive text summarization, however with respect to previous researches done, (Zhang, Xu and Wang, 2019) it is found that transformers outperform most of the other deep learning approaches as of today but there was no much research on optimizing them for a much better performance.

This research will be focused on creating a generalized solution by achieving the optimized transformer architecture from a couple of the top tier existing architectures, via fine-tuning and performing hyperparameter optimization along with handling abstractive text summarization (Liu and Wang, 2021), therefore we are able to maximize the performance of the recommended architecture. The author plans out to make use of generalization where any domain when used the model will be optimizing and adapting towards their respective domain.

1.9.2 Problem Domain Contribution

Neural Networks makes up the backbone of deep learning algorithms which enables them to process complex unstructured data over normal means of machine learning algorithms (Mahajan et al., 2021). It is found that, the need for using advanced deep learning approaches has not been explored in the domain of movie review summarization.

Given that transformers perform well in this field, the proposed solution for this domain will be finding the recommended architecture along with hyper-parameter optimization, to reach its best performance. An additional contribution will be that, the proposed solution will be generalized to any other domain linked with the field of NLP text summarization.

1.10 Research Challenge

The main objective of this research is to achieve the generalized optimal transformer architecture for the field of NLP abstractive text summarization. Transformers were introduced in 2017 by a team at Google Brain and are the most used choice for NLP problems replacing RNN models, given that this architecture was introduced not much longer back brings to a point where there is a lack of research done in the area of transformer optimization for the purpose of abstractive text summarization. (Wolf et al., 2020).

Therefore, creating and finding the recommended transformer architecture along with the optimal parameters which also handles generalization becomes a challenge with very fewer resources to look up to.

Additionally, identifying suitable datasets for this domain (Movie Reviews Summarization) and Generalization is challenging and necessitates a substantial amount of effort in data preprocessing where it is important since we are dealing with NLP and performance optimization related domain.

1.11 Chapter summary

The author outlined the research effort, explained why the research and problem were innovative, and discussed potential challenges that could develop while attempting to tackle them in this chapter. Additionally, the key goals that must be attained for the research to be deemed successful were outlined and connected to the necessary learning outcomes for the degree.

CHAPTER 02. LITERATURE REVIEW

2.1 Chapter Overview

The chapter presents criticisms of previous work on using abstractive text summarization, including deep learning approaches like transformers, for movie review summarization. The author also attempts to create a more generalized model that can handle multiple domains beyond just movies. In the end, the author identifies the best transformer design by fine-tuning the model and adapting it to new data and hyperparameters through domain generalization, which results in the most favorable outcomes.

2.2 Concept Map

The literature review's project scope is depicted in a concept map, and the primary study areas are highlighted as nodes. The concept map was created to ensure that all relevant literature was included, and it can be found in **APPENDIX I**.

2.3 Problem Domain

The simplicity of selling products or services to customers is growing along with the usage of technology and the internet. Sellers utilize customer feedback to better decide how to improve sales and so attain customer satisfaction (Boorugu, Ramesh and Madhavi, 2019). When it comes to movies, people typically find it quite challenging to quickly determine whether a movie meets their demands by reading the reviews, which may occasionally be very lengthy and time-consuming (Khan et al., 2020).

2.3.1 User Reviews

A user/customer review is typically referred to be written feedback from a customer who has used a product or service. Consumers frequently use user ratings and reviews to drive their purchasing decisions. Because the review data is unstructured, it becomes more challenging for consumers to compare and understand lengthier reviews (Lackermair, Kailer and Kanmaz, 2013).

User and customer reviews are extremely important to major corporations like tourism and hospitality as they constitute the primary engine for the country's economic growth and development. where tourists from over the world may blog about their experiences and share their reviews online in numerous formats (Mukherjee et al., 2020).

2.3.2 Corporate Advantage

It is also known that it costs at least five times as much time and money to acquire a new customer as it does to keep an existing one, so it is important to learn how to foster customer loyalty to the brand, business, or service that is being offered. Customer satisfaction is essential to the survival of corporate industries. Understanding client expectations through their feedback or reviews helps business industries grow and fix faults (Pizam and Ellis, 1999).

On the other hand, companies like Netflix or Amazon Prime can use the movie summaries to help users and understand the watching pattern or their interest. Likewise, the movie-related industries need to allow the customers to quickly scan the summary and quickly decide whether they should be watching it or not (Khan et al., 2020).

2.3.3 Text Summarization

With the massive accumulation of information/data on the internet nowadays, it is extremely difficult to extract relevant information from a large number of textual documents. The goal of text summarizing is to provide a condensed yet meaningful version of a lengthy textual content (Shi et al., 2020).

We all know that text summarization has several uses in a variety of internet-based fields, including search engines that are used for querying and e-commerce sites that utilize sentiment analysis to determine client satisfaction with items (Etemad, Abidi and Chhabra, 2021).

However, in the movie industry, consumers may utilize text summarization to simplify customer reviews of movies, which are often lengthy and time-consuming to read. This enables users to make better decisions when they decide whether or not to watch a certain movie (Khan et al., 2020).

2.3.4 Abstractive and Extractive Techniques

Generally, text summarization is classified into two which are; abstractive text summarization and extractive text summarization, however the approach for creating a hybrid model for text summarization is possible (Alsaqer and Sasi, 2017). The abstractive text summarization technique aims to produce the sentences on its own and then uses them to provide a coherent summary. Therefore, the summary's content will vary from the original context yet still convey the same idea (Mahajan et al., 2021). Additionally, it is well recognized that a strong abstractive summary encompasses the input's key details and is linguistically fluent (Zhang et al., 2020).

The extractive text summarizing method focuses on picking out key phrases or groups of phrases from the original input content and combining them to produce a concise yet insightful text summary. It is determined which sentences should be included as parts of the summary based on the statistical and linguistic characteristics of the sentences (Gupta and Lehal, 2010). A hybrid system is one that combines various strategies to produce a single system. However, hybrid text summarizing systems do exist, for instance, using a combination of extractive and abstractive summarization can be utilized to generate a hybrid system that uses encoder-decoders (Kirmani et al., 2019; Abolghasemi, Dadkhah and Tohidi, 2022).

Table 2: Comparison of Text Summarization Techniques (Self-Composed)

Abstractive	Extractive
Paraphrases content like humans do, meaning it creates its own context (Mahajan et al., 2021)	Doesn’t create its own context but uses the best possible phrases from the original document (Gupta and Lehal, 2010)
A vast number of datasets are available to experiment working in this domain.	Capable of visualizing sentence scores and investigating gradient-based ways to calculating the contribution of each input token to score prediction (Pai, 2014)
There is a probability of creating information which may be faulty or that gives a different in meaning compared to the original text.	There is a possibility that the combined sentences made from the extracted sentences will contain errors.

2.3.5 NLP with Deep Learning

NLP is a method for computers to intelligently and effectively analyze, comprehend, and derive meaning from human language, as opposed to other approaches that only focus on the interactions between human language and computers. Deep learning techniques are increasingly being used in the field of AI compared to traditional machine learning approaches due to their success rates in handling difficult high computing learning tasks (Lopez and Kalita, 2017; Mahajan et al., 2021).

In today's NLP, machine learning is prominent, but for the most part it only involves numerically optimizing the weights of characteristics and representations that have been created by humans. Deep learning aims to investigate how computers can utilize data to create features and representations suitable for challenging interpretation tasks (Socher, Bengio and Manning, 2012).

2.3.6 Transformers

Open-source library Transformers contains modern transformer architectures that have been thoroughly developed and are integrated by a common API. Pretraining has enabled the efficient use of this capacity for a wide range of activities, and these designs have permitted the construction of higher-capacity models. Transformers are designed to be easy for practitioners, expandable for researchers, and quick and reliable in industrial deployments (Wolf et al., 2020).

It has been demonstrated that the modern generation of pre-trained language models based on transformers is rather competent at identifying syntactic signals like noun modifiers, possessive pronouns, prepositions, or co-referents, as well as semantic cues like entities and relations (Brasoveanu and Andonie, 2020).

Hugging Face Hub offers a variety of transformer designs, including BERT, GPT2, T5, PEGASUS, and many others. The figure below represents the daily average for unique downloads of the pretrained transformer model architectures between Oct 2019 to May 2020 (Wolf et al., 2020).

Figure 1: Transformer Architecture Downloads Rate (Wolf et al., 2020)

(Etemad, Abidi and Chhabra, 2021) research compares various other researchers approaches taken in order to perform abstractive text summarization, these techniques includes the use of transformers and other neural network approaches such as CNN and LSTM RNN networks. The research comparison table below only includes the approaches of transformers used taken abstractive text summarization.

2.3.7 Hyperparameter Tuning

Finding the ideal collection of parameter values to train an algorithm using in order to build a model relevant to the dataset is known as hyperparameter tuning (Liu and Wang, 2021). The calculation of the performance improvement that may be obtained by changing the value of each of the considered hyperparameters from the original value to the value indicated in the target configuration set by the tuning strategy is where hyperparameters make the biggest contribution to improving algorithm performance (Joy and Selvan, 2022).

There are several hyperparameters that play a significant role in performance enhancement; however, not all of the parameters do so; just a select handful do, for example, learning rate, weight decay, number of epochs, batch size, and warmup ratio. As a result, giving critical hyperparameters a higher priority is crucial (aws.amazon.com, 2022).

Automated framework tools, such as Optuna, an open-source framework for hyperparameter optimization built on the Python programming language, does hyperparameter tweaking. The application of numerous hyperparameter optimization techniques, including Grid Search, Random Search, TPE, and CMA-ES algorithms, was made easier by this framework (Joy and Selvan, 2022).

2.3.8 Generalization

Generalization now plays a significant part in resolving issues in numerous fields that are linked to the same issue. The capacity of a model to generalize to new, previously unobserved data that comes from the same distribution as the model's original data is known as generalization (Neyshabur et al., 2017).

Generalization is a useful strategy for starting with the foundation and improving or specializing in one's field as more unseen domain data becomes available. Therefore, the generalized solution will be able to adapt to even unseen domain data, making this solution to solve a common problem in multiple domain (Zhou et al., 2021).

2.3.9 Data Expansion

The quality of a machine learning or deep learning model depends on a number of factors, one of which is the amount and quality of data fed during model training. There are several approaches to increase or expand your available data one of which is data augmentation (making use of existing data points to create new data points). Making using of new data from the users end by saving as the model is used is another way of exposing new data for model retraining (Shorten and Khoshgoftaar, 2019). To make a generalized model domain-specific, retraining with new data from the specialized domain will be used if needed.

2.4 Existing work

There have been several works done on abstractive text summarization for the field of movie reviews, mainly using the traditional machine learning algorithms. However, there are several limitations which created the need for recent deep learning approaches in order to improve the **system's performance.**

2.4.1 Text Summarization Systems

There were multiple studies done previously in the area of text summarization, regarding both abstractive and extractive text summarization. (Khan et al., 2020) research is related to the domain of movie reviews summarization which is the same as this project domain, where the author has developed an automatic approach to summarize lengthy movie reviews along with the feature where the users are allowed to quickly recognize the positive and negative aspects of the movie with respect to the review processed with. The text summarization approach taken by the author is **extractive approach**, where sentence score ranking plays a major role in creating the summary.

The study of (Boorugu, Ramesh and Madhavi, 2019) is towards the domain of ecommerce but yet related to text summarization for customer reviews on the products they sell, so purpose being that, allowing other customer make better purchasing decisions on products, therefore the hassle of going through all the reviews to making a purchasing decision can be reduced to save time, **abstractive approach** is considered to create the summary, which is a better choice of approach.

The research of (Mukherjee et al., 2020) is another **extractive approach** for text summarization where the author develops a solution for generating personalized aspect-based opinion summaries using a dataset which consists of a large collection of online tourist reviews. In addition, the author has gone a step further to personalize the summary's qualities by using the user's interest. However, using abstractive summarization would be a more effective strategy but also challenging when user interest customization is considered because the sentences have been created using own words rather than with any sentence ranking technique.

(Gupta et al., 2021) research is a comprehensive comparison study with benchmarking results of various pretrained transformer architectures such as BART, BERT, T5, PEGASUS etc... for abstractive text summarization which is an **abstractive approach.** This study includes the various types of datasets been used to explore each model with the evaluations as benchmarking results. The author has also concluded the best performing transformer architecture as **T5** by comparing the evaluation results of the study.

The study conducted by (Mahajan et al., 2021) is also an **abstractive approach** to text summarization with the addition of proper grammar and no repeated words used using a deep learning approach with RNN and likewise (Etemad, Abidi and Chhabra, 2021) research also relates to an experimenting study with various deep learning approaches for abstractive text summarization along with the evaluation benchmarking with a goal in search for the best deep learning approach for the problem.

2.4.2 Algorithmic approaches for Text Summarization

As described in this literature review's Problem Domain section, deep learning approaches are mostly given priority to than traditional machine learning approaches as they can handle highly computational tasks. The author came across of multiple deep learning techniques used aswell as machine learning techniques used for handling abstractive text summarization.

The study of (Khan et al., 2020) starts by first focusing on feature extraction, then transforming reviews into vector spaces, and applying the Naive Bayes machine learning method for review classification utilizing an undirected weighted graph-based ranking algorithm to rank score for each review phrase in graph and then, in order to construct the extractive summary, the highest scoring sentences are selected. However, the author has limited the use of sophisticated deep learning algorithms to improve performance by solely using standard machine learning approaches to tackle the problem.

(Boorugu, Ramesh and Madhavi, 2019) research made use of seq2seq model for text summarization along with the attention mechanism for improved accuracy and the Concept net Number batch word embedding model, which is superior than Glove. Utilizing a 1D convolutional layer, a max pooling layer, an LSTM layer, and finally a fully connected layer at the very end. However, the author's use of generic deep learning algorithms to handle this problem introduces a new constraint that prevents performance from being improved using the most recent deep learning strategy for NLP-related problems, transformers.

The research of (Mukherjee et al., 2020) liked mentioned earlier is an extractive method text summarization based on integer linear programming (**ILP** [Unsupervised method]) to choose an informative subset of opinions centered on the identified aspects. Utilize ROUGE-based criteria to assess and contrast the summaries and get competitive outcomes. Since the dataset is also constrained, extractive summaries could not be particularly insightful; thus, utilizing an abstractive technique might produce superior results, despite the dataset's constrained size.

The study of (Mahajan et al., 2021) focus of the authors' study is utilizing the **encoder-decoder** model with the attention layer to produce text summaries with good syntax and no repeated words. the creation of an encoder-decoder model with gated recurrent units and training it to provide an abstract summary of a piece of writing. Although the author employed deep learning, its application in production required real-time training so that it could be updated with the most recent content over time.

2.4.3 Usage of Transformers

(Gupta et al., 2021) research employed pretrained models such Pipeline BART, BART modified, T5, and PEGASUS to deal with text summarization as a part of the comparison study done. The ROUGE Scores were used as the evaluation measures. During the experiments, the author employed transformer designs; however, the **hyperparameters** used were **default** and might be tuned for a better performance. The constraints consist of concentrating on developing more reliable models that can further expand the method to produce summaries of varying length and applicable for multi-document summarization.

(Etemad, Abidi and Chhabra, 2021) The author explores with deep learning methods in the broad text summarization domain to determine which method—among a collection that includes RNN, CNN, and Transformers—performs best. The author also considers metrics for model evaluations including BLEU and ROUGE, despite using sophisticated deep learning algorithms, the author was unable to undertake **hyperparameter** tuning to improve the method and obtain a better outcome.

2.5 Technological review

There are many applications for text summarization systems today, especially when researching papers. Users may choose from a variety of contexts, such as research paper materials, customer reviews, etc., much more easily by using summaries to comprehend the context and pinpoint the key concepts. Text summarization tools assist researchers in frequently writing an abstract of their findings. With this technique, text summaries may be extracted or abstracted. In contrast to extractive text summarizing, the abstractive text summarization approach creates its own context, which is a far more logical or human-like written language, and can help with problem solving (Barna and Heickal, 2022). Text summarizers may be quite helpful in highlighting the key elements of reviews by providing a summary of user reviews, which can sometimes be very extensive and descriptive.

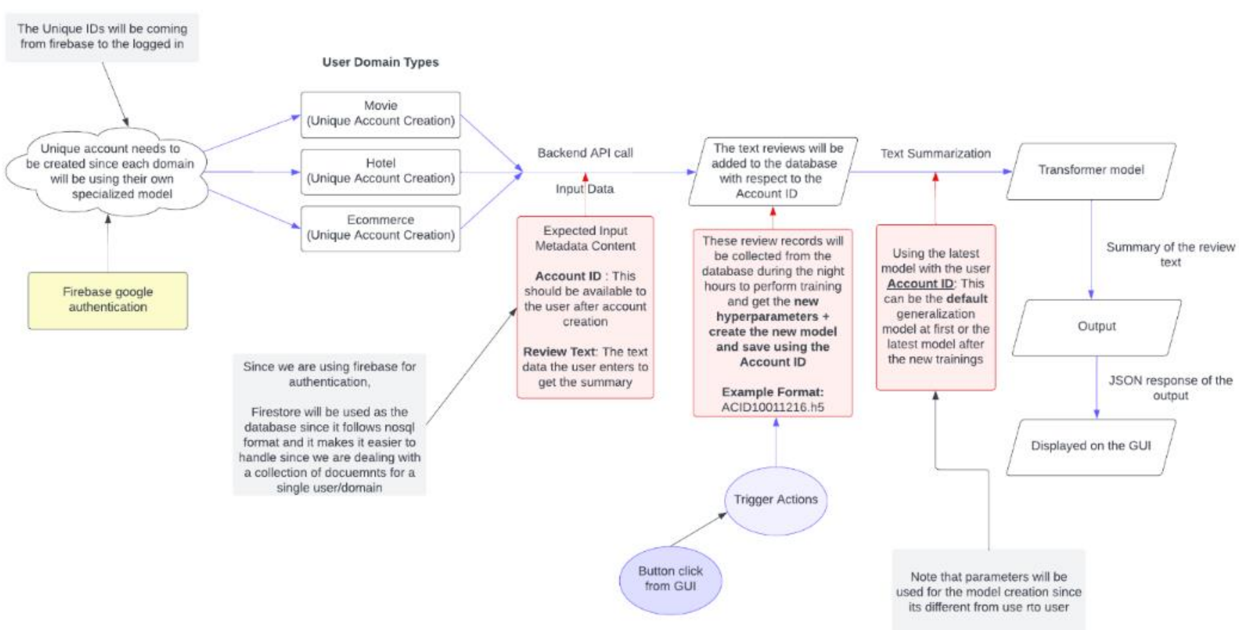
Traditional machine learning and deep learning approaches has been widely used for text summarization for the domain of movies reviews, however advanced deep learning approaches such as Transformers has not been explored for the domain of movie reviews but yet been used in other case scenarios.

Even though traditional machine learning and deep learning approaches performed well, there was a limitation to push the boundaries with new approaches. That's where transformer optimized was considered via repeated hyperparameter tuning with exposure to new data and making this generalized to any domain.

2.5.1 Proposed architecture for the Generalized Text Summarization System.

The provided diagram illustrates a methodology used for developing a generalized system for text summarization. The approach involves improving the performance of the system by retraining the model with relevant domain data and identifying a new set of hyperparameters that are suitable for the updated data. The model is then retrained using the optimal transformer architecture selected, making it more specific to the domain, and thus, enhancing the performance of the system.

Figure 2: Proposed Generalized Abstractive Summarization System Process Flow (Self-Composed) – (*view high qual version*)



2.5.2 Machine Learning Text Summarization Techniques

(Boorugu, Ramesh and Madhavi, 2019) points out a previous research where a system was built that uses a hybrid classifier approach with machine learning algorithm combination of SVM and Naïve Bayes in sync with fuzzy logic and they also concluded that with the increase in the classifier count the accuracy can also be increased. They also made use of supervised ML algorithms such as KNN for the classification of the reviews which then combining appropriate words for identifying the features of the product.

(Khan et al., 2020) proposed system was for the movies domain using the customer reviews, the author broke down proposed methodology into segments of which is preprocessing, feature extraction, review classification and finally review summarization. The Nave Bayes (NB) classification method, which is regarded as a robust classifier and may achieve greater accuracy, was used to categorize the reviews from negative to positive using supervised ML classification technique, It is clear that an extractive summarization approach was used because the text summarization phase was completed in several stages, starting with the creation of a graph from classified reviews, followed by the ranking of graph nodes and the selection of the top rank sentences for the summary generation.

Initially, these machine learning methodologies were given a lot of significance, but as time has progressed on, new technologies and techniques have emerged that can utilize deep learning techniques like RNN, CNN, etc. to perform better.

2.5.3 Deep Learning Text Summarization Techniques

Numerous studies have been conducted on deep learning methods for abstractive text summarization, such as with the usage of CNN, LSTM-CNN, Convolutional Seq2Seq, Sequence to Sequence RNN, Convolutional Sequence to Sequence, Transformers, T5, BART, BERT etc.… which were trained on a general dataset such as from Gigaword, DUC 2002, DUC 2004, CNN Daily Mail, DUC, Xsum, Newsroom such datasets, in order to get an evaluation comparison on which outperforms the rest and eventually the T5 Transformer outperformed the rest of the other techniques in the case of abstractive text summarization (Etemad, Abidi and Chhabra, 2021).

(Shi et al., 2020) has conducted a thorough analysis of latest developments in seq2seq models for the task of abstractive text summarizing. The author's analysis includes a full review of several distinct seq2seq models for abstractive summarization.

Out of which transformers are the advanced deep learning approach for text summarization which is an encoder-decoder model with attention layer which helps it to generate better results than a traditional simple RNN architecture (Mahajan et al., 2021).

2.5.4 Available Datasets for generalized text summarization

There are two datasets that the author will be exploring throughout the development of this project. One of which is the Amazon movie reviews dataset from Stanford University Education, which contains data within the span period of more than 10 years including 8 million review data records (McAuley and Leskovec, 2013).

This dataset will be used to test out the solution for the problem domain which is abstractive text summarization for movies. Given that the author is able to create the solution for the domain of movies then, the author then plans to generalize the solution using another dataset named as Gigaword which is from TensorFlow datasets which was used previously for creating generalized content for text summarization (Kouris, Alexandridis and Stafylopatis, 2019).

2.5.5 Available Datasets for Generalized Text Summarization

There are two datasets that the author will be exploring throughout the development of this project. One of which is the Amazon movie reviews dataset from Stanford University Education, which contains data within the span period of more than 10 years including 8 million review data records (McAuley and Leskovec, 2013).

This dataset will be used to test out the solution for the problem domain which is abstractive text summarization for movies. Given that the author is able to create the solution for the domain of movies then, the author then plans to generalize the solution using another dataset named as Gigaword which is from TensorFlow datasets which was used previously for creating generalized content for text summarization (Kouris, Alexandridis and Stafylopatis, 2019).

2.5.6 Preprocessing Techniques used in Text Summarization

Text preprocessing is very important when it comes to dealing with text related data. In earlier studies, a variety of text preprocessing approaches were utilized for text summarization.

Sentence segmentation is a fundamental step in NLP applications including IR, machine translation, semantic role labeling, and summarization. It is the process of identifying boundaries within a document that divides the document's text into sentences, typically from a strong point of punctuation like (full stop, explanation mark, question mark, etc.), Tokenization and stop words removal will then be performed. Tokenization will be carried out by the tokenizer program to split the sentences into distinct words by splitting them at whitespaces such as blanks, tabs, and any strong punctuation. Stop word removal is also used to remove frequently used words in the document such as "I," "an," and "a" because these words carry little meaning and are best removed from the document (Khan et al., 2020).

Other researchers have incorporated a variety of other techniques, including noise removal, which eliminates unnecessary text from the input document, such as the header and footer, and named entity recognition (NER), which recognizes words in the input text as names of things like people, places, and things, among others (Barna and Heickal, 2022).

Datasets may also contain unwanted records, null records, or redundant records that are absolutely useless. These records or rows with null values are eliminated, unnecessary HTML tags and URL links are also filtered off from the text as a part of text preprocessing. Contraction mapping is crucial and this will be handling which are converting short word formats into longer such as “aren’t” into “are not”. Converting the entire text content into a single case most preferably to lowercase, therefore further character filtration would become very simpler (Mahajan et al., 2021).

2.6 Evaluation Techniques

2.6.1 Evaluation Approaches

A machine learning model's performance, as well as its advantages and disadvantages, are understood through the process of model evaluation, which employs many evaluation measures. During the early stages of research, it's critical to evaluate models to determine their efficiency.

The table below shows the available measure and the metrics that can been used to **quantitatively** evaluate the text summarization system.

Table 3: Evaluation metrics for abstractive text summarization (Self-Composed)

Measure	Description	Objective Orientation
Metric: 01		
ROUGE	ROUGE also known as Recall-Oriented Understudy for Gisting Evaluation. Measures are made by comparison between an automatically generated summary/translation against a group of reference summaries (generally human created summaries) (Lin, 2004). ROUGE measures the **recall,** (according to how frequently the terms from the summaries created by humans appeared in those computer - generated.)	Positively oriented. Higher, the better
Equations		
Rouge-1		
Refers to the system summary and reference summary's overlap of unigrams (one-word sequence). $$ROUGE-1 = \frac{(\sum_{Reference\ Summary} \sum_{unigram} Count_{match}(unigram))}{(\sum_{Reference\ Summary} \sum_{unigram} Count(unigram))}$$		
Rouge-2		
Refers to the bigram (two-word-sequence) overlap between the system and the reference summaries $$ROUGE-2 = \frac{\sum_{s\in\{RefSummaries\}} \sum_{bigrams\ i\in S} \min(count(i,X), count(i,S))}{\sum_{s\in\{RefSummaries\}} \sum_{bigrams\ i\in S} count(i,S)}$$		
Rouge-L		
Measures the longest matching word sequence $$ROUGE-L_{(candidate,references)} = max_k\{ROUGE-L_{single}(candidate, references_k)\}$$		

Metric: 02		
BLEU	BLEU also known as Bilingual Evaluation Understudy is a metric used for evaluation for the quality of machine generated text by comparing it with a reference text that is supposed to be generated. (Steinberger and Jezek, 2009). BLEU measures the precision (as to how much words in the generated summaries appeared in the human generated summaries)	Positively oriented. Higher, the better
Equations		

$$BLEU = \frac{Number\ of\ words\ in\ the\ summary\ which\ are\ in\ gold\ standard}{Total\ number\ of\ words\ in\ the\ summary}$$

Different versions of ROUGE exist, including ROUGE-1, ROUGE-N, ROUGE-L, and ROUGE-S. For example, ROUGE-L considers the longest common sequence, whereas ROUGE-S and ROUGE-SU consider skip sequences. (Etemad, Abidi and Chhabra, 2021). Out of which ROUGE-1, ROUGE-2 and ROUGE-L is considered as the least number ways to get a proper evaluation of the model and the scores lies between 0 to 1. Higher the score, the better.

(Steinberger and Jezek, 2009) Out of both of these evaluation metrics ROUGE score demonstrates the best performance for text summarization as compared to BLEU. (Lin, 2004) claims that they introduced ROUGE, an evaluation package for summarization, and carried out thorough evaluations of the automated measures present in the ROUGE package using three years' worth of DUC data.

2.6.2 Benchmarking

The table given below is the benchmarking results of training transformers with generalized datasets for abstractive text summarization. The author will be also following the same measures for the evaluation benchmarking for the prototype, so it can be comparable.

Table 4: Comparison table for abstractive text summarization using transformers (Etemad, Abidi and Chhabra, 2021)

Researcher	Year	Type of model	Rouge 1	Rouge 2	Rouge L	Dataset
Haoyu Zhang et al.	2019	Transformer with BERT	41.71	19.49	38.79	CNN-Daily Mail
Andrew Hoang et al	2019	Transformer	39.01 36.73 40.87	17.87 14.93 28.59	36.17 29.66 37.62	CNN Daily Mail Xsum Newsroom
Kaiqiang Song et al.	2019	Transformer	40.89 45.93	19.11 24.14	37.60 42.51	Gigaword, Newsroom
Mike Lewis et al.	2019	BART	44.16 45.14	21.28 22.27	40.90 37.25	CNN Daily Mail Xsum
Itsumi Saito et al.	2020	RoBERTa Base	45.80 45.42	22.53 22.13	42.48 36.92	CNN Daily Mail Xsum
Beliz Gunel et al.	2020	Transformer XL	34.273	13.018	32.048	CNN Daily Mail
Colin Raffel et al.	-	T5	43.52	21.55	40.69	CNN Daily Mail

2.7 Chapter summary

The problem, technology domains, prior work, and assessment strategies were all broken down in a concept map at the beginning of this chapter. Then, these four areas were further divided into subtopics and examined based on the research and ideas presented in earlier works of literature. A critical analysis of all the literature has been conducted, contrasting the likes and dislikes of earlier research, potential future work described in the literature, and unique approaches that the author of this study proposes as potentials not before explored.

CHAPTER 03. METHODOLOGY

3.1 Chapter overview

To make sure the research process runs smoothly, it's important to have a clear definition of the methodology used. In this chapter, the author explains the selected methodologies and provides detailed explanations for their choices. Additionally, the chapter presents the project's needs, potential risks, plans to mitigate those risks, timeline, a plan for dividing work, and the expected results.

3.2 Research methodology

When determining the quality of a project, there are a number of important factors to consider, including the cost incurred, the amount of time required, and the weight given to the project's scope. These factors must be effectively managed throughout the project's lifespan, which is when methodologies are required.

The table listed below are the chosen methodologies for the project, where Saunders Researched Onion model has been used (Saunders, Lewis and Thornhill, 2007).

Table 5: Research Methodology (Self-Composed)

Research Philosophy	The author will explore and experiment with numerous techniques as part of a combined strategy to determine which is most effective for reaching the research aim, therefore the **pragmatism** approach was chosen among the positivism, pragmatism, realism, and interpretivism approaches.
Research Approach	This research experiments with several approaches to figure out the best, the **deductive approach** was taken into consideration this was because the research aims at applying a combination of existing model architectures to fine-tune and get the best. As the data analysis **qualitative method** were chosen.

Research Strategy	This area focuses on data collection with respect to the research questions created. **Survey and experiments** were the strategies considered to address the research questions. Both of these strategies are expected as an approach for the **quantitative result** at evaluation.
Research Choice	Weather the research is concerned with qualitative or quantitative aspects depends on the choice of methodology. Even though we ultimately prioritize quantitative findings mainly, **multi-method** was taken into consideration for this study. This is partly because determining the qualitativeness of the data utilized for development is important since, in the end, it will influence the quantitative outcomes.
Time Horizons	**Cross-sectional** will be used since only during the requirement engineering and evaluation phase the data will be gathered and therefore not repeatedly collection over time.
Techniques and procedures	Here, data collecting and analysis methods are considered. We'll utilize sources including internet news, discussions, reports, surveys, publications and organizational records.

3.3 Development methodology

3.3.1 Life cycle model

The project's research development methodology of choice was the **Agile** Software Development Life Cycle. This is a result of the project's reliance on an iterative development method.

3.3.2 Requirement elicitation methodology

Conducting Surveys via questionnaires, review more previous research done, experimenting with various transformer architectures and brainstorming will be the approaches taken in-order to communicate and gather **insights** for the projects need.

3.3.3 Design methodology

The system's simplicity of expansion and future growth was considered by the author to support incremental methodology, therefore **SSADM (Structured Systems Analysis and Design Method)** was chosen as the Design Methodology for the project.

3.3.4 Software development methodology

Functional programming methodology will be used for the development methodology for the project, this is due to the project's ease of future developer enhancement, making it simpler.

3.4 Project management methodology

Prince2 is the chosen approach for project management. This aids the author's ability to be extremely flexible as well as operate in controlled environments.

3.4.1 Schedule

3.4.1.1 Gantt chart

The project’s schedule is presented as a Gantt chart.

(Gantt chart is present in the next page)

P.T.O

Figure 3: Gantt Chart (Self-Composed)

3.4.1.2 Deliverables

The deliverables and respective dates are specified in the table below.

Table 6: Deliverables & Dates (Self-Composed)

Deliverable	Date
Literature Review A thorough examination and evaluation of existing research and proposed solutions.	27th October 2022

Project Proposal Document + Ethics Forms Initial proposal of the project.	9th November 2022
Software Requirement Specification Documentation outlining the requirements that must be met, designed as the ultimate prototype, including data collection methods.	24th November 2022
Proof of Concept with Implementation Presentation Performing a presentation regarding the implementation along with the proof of concept	23rd December 2022
Project Specifications Design & Prototype A functional prototype with all its main features included as stated. Along with a documentation of the design approach followed.	16th February 2023
Test & Evaluation Report Documented Evaluation Report conducted on the Prototype.	23rd March 2023
Draft Project Reports A draft thesis submission, in order to get supervisors feedback	10th April 2023
Final Thesis Final report detailing the research and project decisions	27th April 2023
Review Research Paper A review paper reviewing published existing systems in handling abstractive text summarization.	8th April 2023
Final Research Paper A research paper about the experimentations done with the transformers hyperparameters.	25th April 2023
Public project repository A publicly accessible project repository to setup and test the development	27th April 2023

3.5 Resources

3.5.1 Software requirements

- ***Operating System* –** Microsoft Windows OS will be used for the research, documentation and for the complete project implementation (end to end), due to its flexibility compared to another operating system (Najmuddin et al., 1970).
- ***Python* –** Machine learning & Deep learning model development and APIs creation to serve the models and handle logic will be implemented by using the Python language. Python is a general-purpose language that has been used most widely in data-science related projects and in backend frameworks link Flask and Django (McKinney, 2017).
- ***Flask* –** Backend web framework for API development for the prototype. This will be used to access/transfer data to and from the data science models developed, due to its lightweight and flexibility (Chauhan et al., 1970).
- ***Torch* –** Libraries that will be used during the development of the data science models, since lightweight and ease of use compared to TensorFlow, building models will be quicker (Paszke et al., 1970).
- ***Jupyter Notebook & Google Colab* –** Used for Machine-learning/Deep learning model development in this project, it's an Integrated development environment for programming, where google colab provides on the go resources to use for model training (Canesche et al., 1970).
- ***TypeScript (React)* –** JavaScript framework which is used for the development of the frontend application interface of the project. Here is where the user will be able to input and view their data.
- ***Vscode* –** The project's development environment. This will be utilized while creating the codebase for the backend API and frontend development.
- ***Zotero* –** Referencing software that keeps a copy of all the articles as well as managing the references for research papers
- ***MS Office/ Google Docs/ Figma* –** Software & tools which will be used to create figures, reports and handle documentations.
- ***Google Drive/ GitHub* –** Backup platform and code management system to help keep backup of all documents and code.

- ***Git*** **–** Version control system which will be used to keep track of the changes made in the project code and manage code changes.
- ***Firebase*** **–** Application development platform which helps to build and grow apps, its also known as the Backend as a Service.

3.5.2 Hardware requirements

- **Core i5x Processor (8th generation) or above –** Above average processing power required to perform high resource intensive tasks (such as model training).
- **Nvidia MX130 GPU or above –** To handle data science model training processes.
- **16GB RAM or above –** Sufficient amount of RAM needed to run multiple applications (client + server), model training also consumes a lot of CPU and RAM.
- **Disk space of 30GB or above –** To store project data and applications.

3.5.3 Technical skills

- Good understanding about machine learning and deep learning concepts.
- Good understanding about Natural Language Processing and its data preprocessing methods.
- Good understanding about transformers and how to work with hyperparameters in general along with the knowledge of its use.
- Research writing skills

3.5.4 Data requirements

- Amazon Movie review data – From Stanford University Education.
- Gigaword, Xsum & CNN Daily News dataset – From TensorFlow datasets which will be used for generalization model.

3.6 Risks & mitigation

The table given below defines the possible risks which can be encountered during the process of the project development along with the possible mitigation steps.

Table 7: Risk Management Plan (Self-Composed)

Risk	Magnitude	Frequency	Mitigation Plan
Losing the development project codebase/repository	5	4	Using GitHub and external backup to keep a latest copy of the project codebase.
Personal computing breaks down during the project timeline progress.	5	4	Upload the complete backup to GitHub and Google Drive, use University Lab service to continue project work, till personal machine recovery.
Unable to complete all mentioned project deliverables on time	4	4	Prioritize and create a timeline to complete the deliverables.
Project documentation corruption	5	3	Use a dedicated folder under the same GitHub repository and push all latest documentation changes & use cloud-based documentation approach
Insufficient knowledge on the project domain	5	3	Performing an intensive research on the problem domain along with the research domain.
Any unavoidable personal health risk – Sickness	3	1	Create weekly goals to complete and keep them updated.

3.7 Chapter summary

The chapter describes the methodology used for a project, including research and development methods, project management strategies, and the rationale for those choices. It also outlines project requirements, work division plan, expected outcomes, and potential risks with their corresponding mitigation strategies.

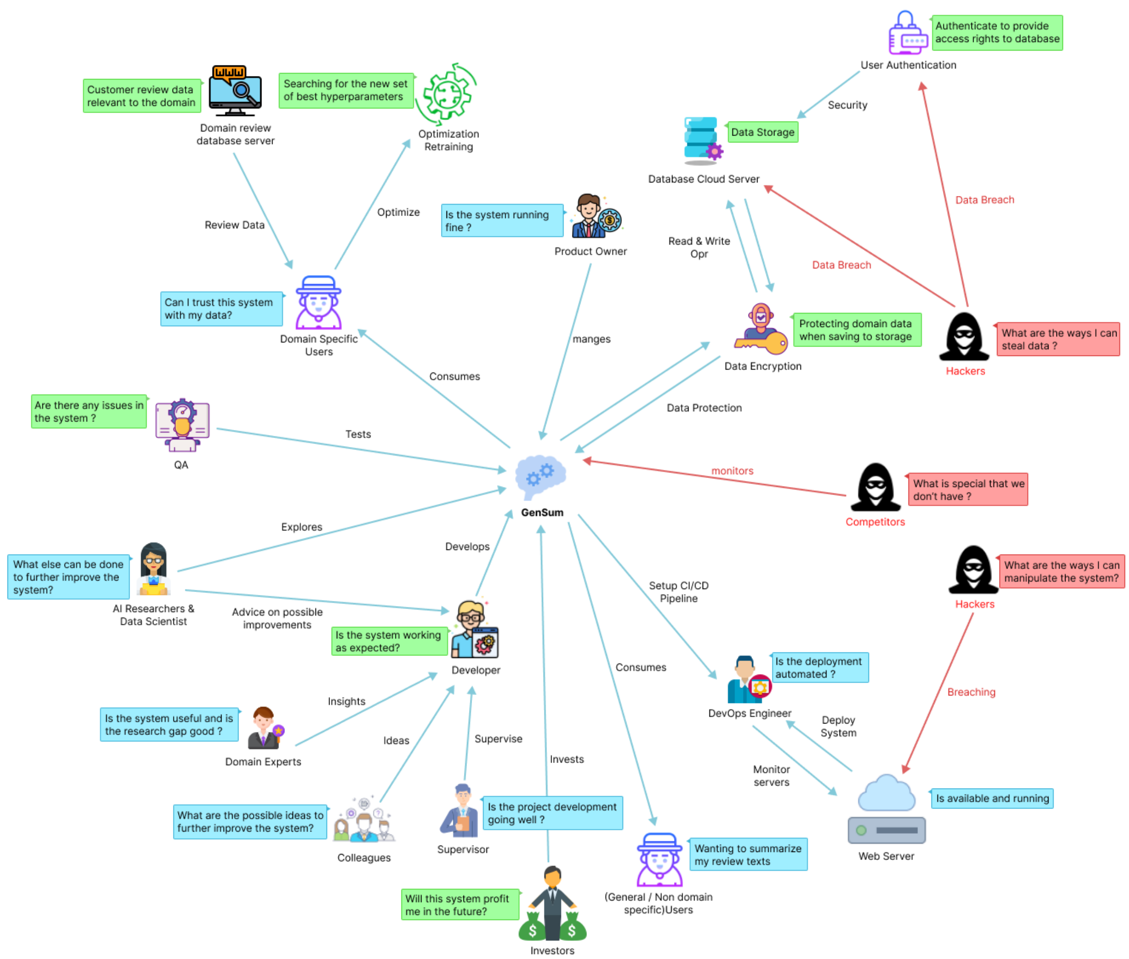
CHAPTER 04. SOFTWARE REQUIREMENTS SPECIFICATION

4.1 Chapter overview

This chapter explains the process of identifying and gathering essential needs from stakeholders. The author discusses the use of a rich picture diagram and a stakeholder onion model to record stakeholder engagement, interaction points, and individual responsibilities. The chapter also covers the techniques used for requirement gathering, including creating functional and non-functional requirements, use case diagrams, and prototypes based on the gathered results.

4.2 Rich picture

Figure 4: Rich Picture Diagram (Self-Composed) – *(view high qual version)*



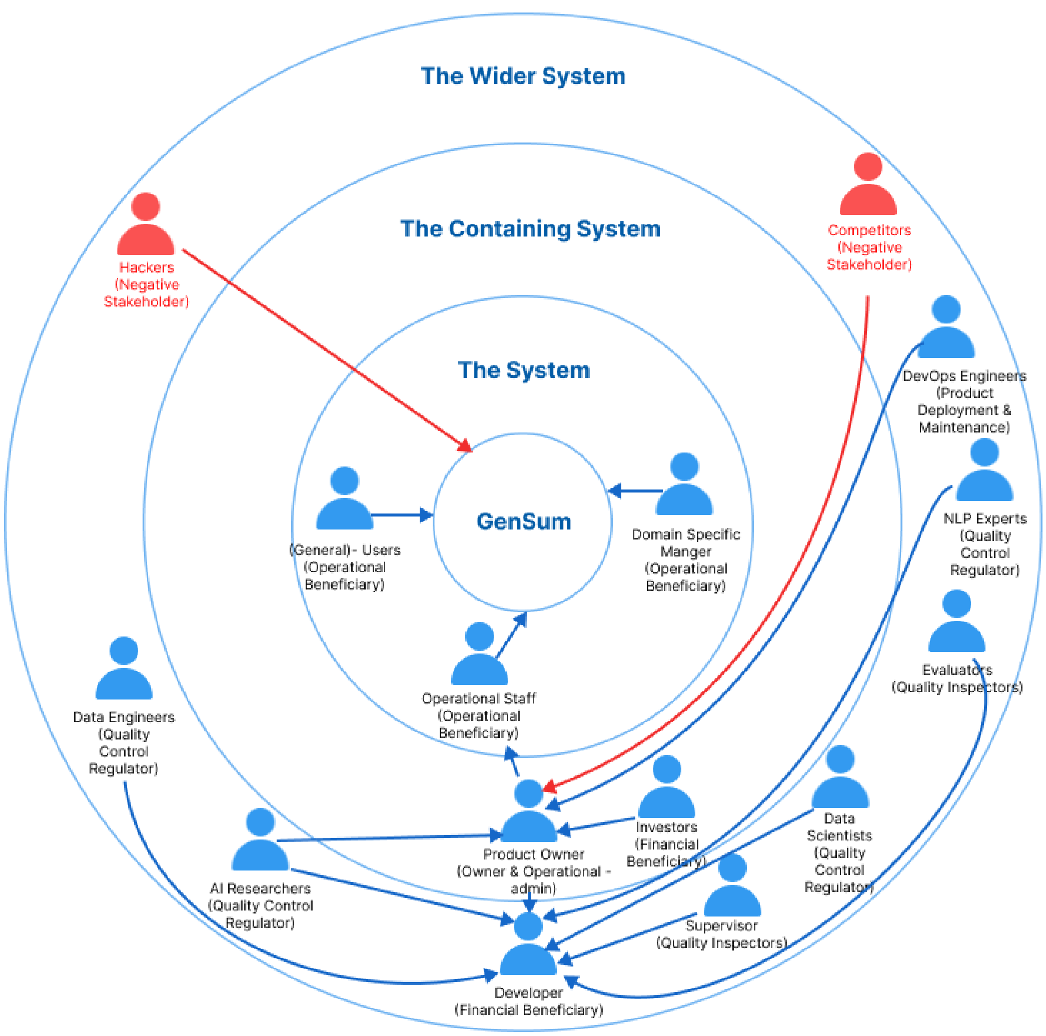
The diagram above depicts a bird's-eye view of the surrounding region, as well as how certain stakeholders might interact with the system and profit from it. Along with the knowledge gained by the researcher to improve the system, the potential negative impacts on the design and prospective critical analyses are also identified.

4.3 Stakeholder analysis

The following section identifies the important stakeholders involved in the system, their relationships, and their roles. This information is presented using the stakeholder onion model, and their perspectives are further elaborated upon.

4.3.1 Stakeholder onion model

Figure 5: Stakeholder Onion Model (Self-Composed) – *(view high qual version)*



4.3.2 Stakeholder viewpoints

The table below provides information about the stakeholders, their functions, and the actions related to them.

Table 8: Stakeholder Viewpoints & Requirements (Self-Composed)

Stakeholder	Role	Benefits/Description
Developer	Functional beneficiary	Works on developing the system
Investors		Profit is generated through system investment and money from marketing and user subscriptions.
Product Owner	Owner & Operational admin	Owns the system and has control over the system
Data Scientists	Quality Control Regulator	Provides performance enhancements for the models and algorithms used in data science.
Data Engineers		Gives guidance on potential data that may be used to generate the best suggestions possible.
AI Researchers		Conduct research in the specified area to enhance and implement reliable text summarizing models.
NLP Experts		Offers specialized guidance and insights on the field knowledge, to enhance the functionality of the system.
Domain Specific Manager	Operational Beneficiary	Text reviews are used as inputs for abstractive summarization, and the model is retrained with prior inputs as new data to increase performance.
General Users		Unless specifically assigned or retrained, typical users will utilize a general abstractive summarization model.
Operational Staff		Ensures that the system is up and functioning while responding to user requests and problems.
DevOps Engineers	Product Deployment & Maintenance	Makes ensuring the system is up and running in the cloud and is serving users without being throttled

Hackers	Negative Stakeholder	May manipulate the review data stored in the database which will affect the retraining process.
Competitors		May build competing systems that may outperform the existing system.
Evaluators	Quality Inspector	Checks to see if the system is ready for production use and puts it through its paces.
Supervisor		Checks to see if the system development is progressing well without any issues.

4.4 Selection of Requirement Elicitation Methodologies

There were several requirement elicitation approaches used to collect needs for the creation of the research project. The approaches selected for this were literature review, survey, interviews, prototyping, brainstorming and self-evaluation. The following is a discussion of the rationales behind selecting the mentioned requirement elicitation approaches.

Table 9: Requirement Elicitation Methodologies (Self-Composed)

Method	**Description**
Literature Review	To determine research gaps in the chosen domain of interest and the intended topic of study at the project's outset, the author conducted a thorough literature analysis. Current systems were researched together with comparable technologies that might be applied to the existing systems that were referenced in literature in order to discover research gaps available in technologies that can be used.
Survey	A questionnaire was utilized as a survey instrument to obtain requirements and opinions from possible users of the suggested system. The author will benefit from this sort of poll in understanding people's thought processes and expectations for the prototype. It will also enable the author to explain whether or not the targeted users will benefit from the suggested solution.
Interviews	Interviews were performed to gain expert insight into domain-specific requirements and to determine the best method to address the issue at hand

	while adding to the body of knowledge through research. Interviews were determined to be the greatest source of information because the field is new and the technical expertise needed is very precise. Additionally, this technique allowed for the qualitative evaluation of the suggested system, allowing for the identification of any shortcomings or difficulties that could need to be resolved during prototyping.
Prototyping	The project was chosen to follow the Agile Software Development Life-cycle, thus prototyping would allow the author to test and evaluate the prototype while iteratively trying out several alternative implementations to find any potential areas for improvement.
Brainstorming	Whether you're attempting to come up with a broad subject before you start your research, you're trying to focus more specifically, or you're determining what evidence to use for a particular paragraph, brainstorming is a useful technique to produce ideas at every step of the process. In order to assess the system for personally, the author has a number of brainstorming sessions with his colleagues at various project stages.
Self-Evaluation	Self-evaluation is done in order to examine the currently available applications, do competitor analyses on the current systems, and get insight into how negative stakeholders, such as hackers, can breach the system and find a way around to protect the data and the system.

4.5 Discussion of Findings

The relevant key stakeholders are split up into groups where the chosen best methodology was used for each group. **APPENDIX C.1** contains a complete breakdown of these stakeholders.

4.5.1 Literature review

Table 10: Literature Review Findings (Self-Composed)

Discussion of Findings	Citation
In the completion of the literature review on the existing work done, it was identified that abstractive text summarization systems for customer	(Boorugu, Ramesh and Madhavi, 2019)

reviews helps users to make better and quicker decisions on their actions let it be on buying products or watching a movie, user review summarization proves to save time for customers.	
When exploring technologies that can be applied to achieve the required outcome, it was clear that traditional machine learning and deep learning approaches were only used for abstractive text summarization in the domain of movie reviews. Leaving the usage of advanced deep learning approaches such as Transformers untouched for this domain.	(Khan et al., 2020)
It was identified that transformer optimization has not been looked into when working with transformers in abstractive text summarization domain in general and not specific to the movie domain.	(Gupta et al., 2021)
Dataset related to working with model generalized has been used previously and is recommended to be used if researchers are willing to work with the idea of generalization for the domain of abstractive text summarization.	(Kouris, Alexandridis and Stafylopatis, 2019)

4.5.2 Brainstorming

The author engaged in brainstorming across various project phases. These were carried out both with the authors' colleagues and supervisors as well as through a self-analysis process.

Table 11: Observations Findings (Self-Composed)

Criteria	**Discussion of Findings**
Able to figure out several other research gaps/ limitations which can be fit into the current project domain in order to increase the magnitude of research effort.	Multiple ideas were brought up as the result of the brainstorming session. The concept of creating a performance adaptive generalization model was brought up by the authors supervisor, along with several other approaches to increase the performance of the system exponentially such like making use of the new data from the domain users for retraining and combine all data with the common domain for retraining since the data count increases with respect to the common domain user.

4.5.3 Interviews

Interviews with experts and researchers in the relevant domains were performed to obtain insights on the technical domain competence. To determine the project requirements, experts and researchers in ML and abstractive text summarizing systems were chosen. 2 PhD candidate in ML and Computational Linguistics, 1 NLP Researcher, 2 Software Architects, 1 Software Engineer and 1 Lecturer with MSc completion were interviewed. The interview outcomes were processed to a thematic analysis based on the following themes. Interview participant details can be found at over **APPENDIX C.2**

Table 12: Interview Thematic Analysis (Self-Composed)

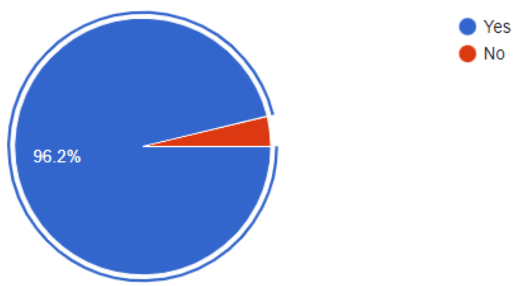
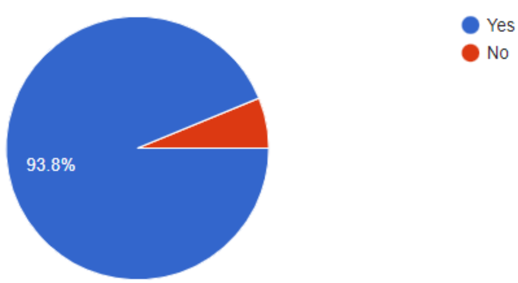
Code	**Theme**
Data handling	Data Collection & Data Preprocessing
Transformer architectures	Best performing transformer architectures
Generalization	Handling adaptive generalization
Research scope	Research gap and scope
Hyperparameter tuning	Automatic hyperparameter tuning & model retraining
Hybrid transformers	Looking into hybrid transformer combinations
Custom transformers	Customizing the transformer architecture
Prototype	Prototype features and suggestions
Business benefits	Understanding which and how businesses would benefit
Evaluations	Understanding the importance and evaluation ways

Theme	**Conclusion**
Data handling	Data accessibility and data preparation techniques are crucial considerations for a data science project. Since every domain would initially employ the same model, PhD candidates proposed utilizing validated and well-researched datasets for the field of generalization to ensure the quality of data. Since text data may contain characters from other languages unless the project is restricted to English language support alone, NLP experts raised worry about the language of the text included in the project scope.

Transformer architectures	The interviewees stated that NLP tasks like text summarization and sentiment analysis may be successfully solved using transformer designs like BERT, GPT-2, Roberta, T5, and others. They advised using the most recent version of these models because they are frequently upgraded and developed in new, improved forms. To keep track of these upgrades, they advised examining daily analytics from websites like Hugging Face, such as download and like counts.
Generalization	Due to their scalability and performance advantages, the software engineers and architects suggested employing NoSQL databases like MongoDB or Firebase for storing data related to domain specific managers.
Research scope	Experts agree that using optimized transformers to solve the issue is a brilliant idea, but given the project's time constraints, they advise prioritizing the movie domain and postponing the task of developing a broader adaptive solution.
Hyperparameter tuning	The NLP researchers and Lectures suggested several ways of using tools and libraries to help with hyperparameter tuning since doing this manually is very time consuming and unnecessary effort.
Hybrid transformers	The hybrid transformer combination with ensemble techniques was well-loved by PhD applicants, but many believed the project scope was growing too large and risky for the time available.
Prototype	Interviewees are interested in the novel domain-specific retraining system and recommended incorporating a pretrained model for sentiment analysis if time allows.
Business benefits	Most of the interviewees suggested the Movie domain, Tourism, Ecommerce, Book, Researchers would find this useful in summarizing their customer reviews on their businesses.
Evaluations	PhD candidates and NLP experts emphasized the need for evaluations in the adaptive generalization model and suggested limiting the project to a maximum of three domains for easier comparison and clearer demonstration

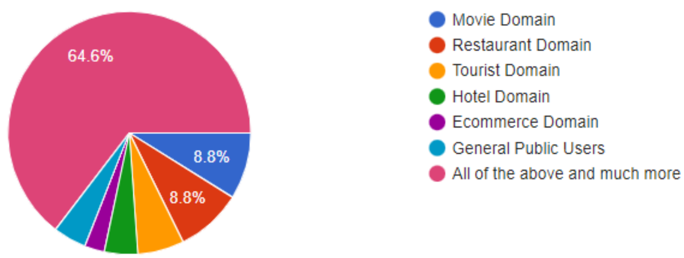
4.5.4 Survey

Table 13: Survey Analysis (Self-Composed)

<table>
<tr><td>Question</td><td>Have you ever realized that reading lengthy reviews takes a significant amount of time?</td></tr>
<tr><td>Aim of question</td><td>To determine whether the audience as a whole considers reading lengthy reviews to be a time-consuming activity.</td></tr>
<tr><td colspan="2">Findings & Conclusion
It can be concluded that a large part of the audience (more than 90% of the audience) finds that’s reading lengthy reviews is a time-consuming hassle which also proves that they would appreciate if there would be a quicker approach for this problem, like a summarization. This also concludes to see a positive correlation from the results which was expected from the author of the project.

</td></tr>
<tr><td>Question</td><td>Do you believe that developing a generic system for all domains would be a wise course of action?</td></tr>
<tr><td>Aim of question</td><td>Ensuring that developing a generic system would be beneficial in all domains</td></tr>
<tr><td colspan="2">Findings & Conclusion
It can be concluded that most of the participants (more than 90% of the audience) agrees that developing a generalized system which can adapt to the domain as they use, is beneficial and worth the effort to process with the project research. This also concludes to see a positive correlation from the results which was expected from the author of the project

</td></tr>
<tr><td>Question</td><td>Who do you think will most benefit from this system?</td></tr>
<tr><td>Aim of question</td><td>Getting to know about the thoughts of the participants about to whom the system would mostly benefit from?</td></tr>
</table>

Findings & Conclusion

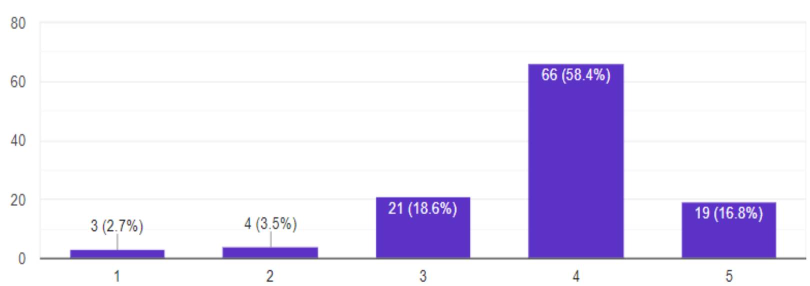
It can be concluded that a majority of the participants (more than 60%) finds that this system will benefit the movie, restaurant, tourist, hotel, ecommerce domains (these domains were considered since they are mostly interacted with the users on a daily bases and uses customer reviews for their domain as a part of their business) as well as the general users.



Question	How much do you think that this system would benefit you?
Aim of question	Getting to know how much the system would benefit the general participants which are NOT domain specific

Findings & Conclusion

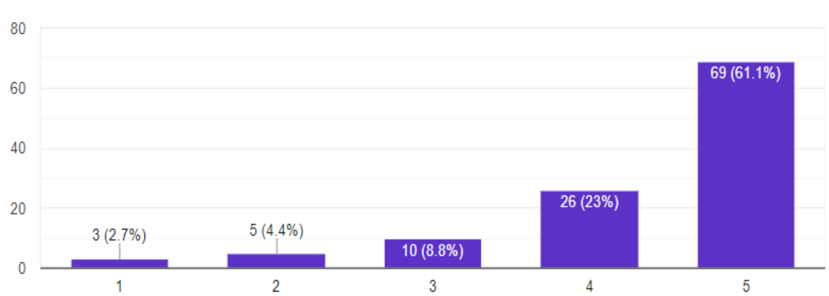
From the statistics graph, it can be concluded that roughly 75% of the audience finds that the system would benefit them for their general work or needs given that it's not domain specific to them, which is a positively correlated result from the achieved statistics.

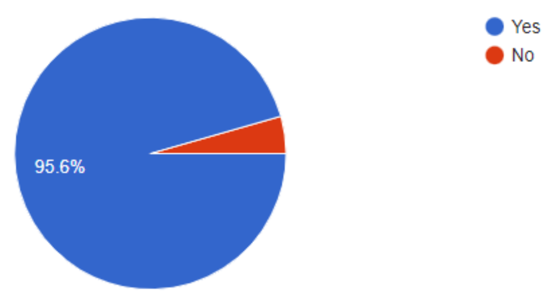


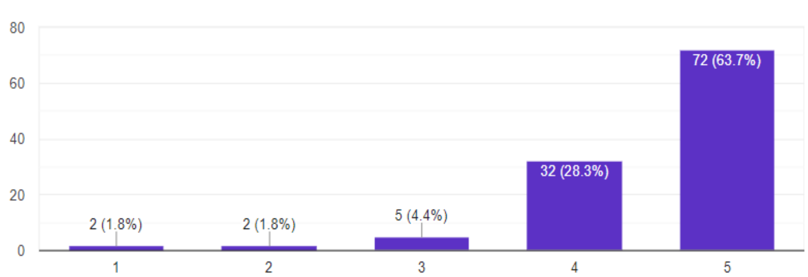
Question	How much do you think that this system would benefit businesses?
Aim of question	Getting to know from the participants as to how much the system would benefit businesses/domains in solving this problem.

Findings & Conclusion

From the statistics graph, it can be concluded that roughly 84% of the audience finds that the system would benefit the businesses, which is a



<table>
<tr><td colspan="2">positively correlated results from the achieved statistics and that's what the author expected to achieve.</td></tr>
<tr><td>Question</td><td>Before making a reservation or booking a movie or a hotel, do you read the customer reviews?</td></tr>
<tr><td>Aim of question</td><td>Getting an idea from the audience if in general they give importance to customer/user reviews to any domain before consuming their product or services.</td></tr>
<tr><td colspan="2">Findings & Conclusion


It can be concluded that a majority of the participants (more than 95% of the audience) agrees that they value and read customer reviews before they consume one's product or service. Therefore, making customer reviews a major contributing factor for business growth.</td></tr>
<tr><td>Question</td><td>How much you think customer reviews are important with respect to any domain?</td></tr>
<tr><td>Aim of question</td><td>Getting an idea from the audience to see how much they value customer reviews.</td></tr>
<tr><td colspan="2">Findings & Conclusion
From the statistics graph, it can be concluded that roughly 90% of the audience finds that customer/user reviews are very important irrelevant to the domain, which is a positively correlated results from the achieved statistics and that's what the author expected to achieve.

</td></tr>
<tr><td>Question</td><td>Which additional features would you want to see in this system.</td></tr>
<tr><td>Aim of question</td><td>To identify the systems non-functional requirements which could potentially improve the system.</td></tr>
</table>

Findings & Conclusion
The majority of participant responses were concerned with classifying the review text's sentiment after it had been summarized and with managing a list of review uploads so as to add filtering for the summarized review text based on the sentiment, whether it was positive or negative, along with the sentiment score.

Table 14: Survey Thematic Analysis (Self-Composed)

Code	**Theme**
Convenience	User-friendly
Adjustability	Flexibility
Insights	Reliability
Advancement	Future Enhancement

Theme	**Conclusion**
User-friendly	A group of participants required to upload more than one review and a time/bulk at once.
Flexibility	A majority of the participants requested for sentiment of the summary and the sentiment score to be also included with the output.
Reliability	A majority of the participants found the system being useful in the industry with time.
Future Enhancement	Few participants requested integrate a recommendation system for a specific domain.

4.5.5 Self Evaluation

Comparing similar products from competitors and existing products gives the author an idea of making the project more unique and distinguish new approaches to solve the problem (**Competitor Analysis**). The author will also self-evaluate as to what data needs to be protected and how from the hackers. Few of the abstractive text summarization tools which are out there are listed and is given below.

Table 15 Competitor Analysis (Self-Composed)

Competitor Analysis Table					
Tools / **Feature**	Summarize Bot	Resoomer	Smmry	Text Compactor	**GenSum**
Summarizing Text	✓	✓	✓	✓	✓
Domain Specific Generalization	✗	✗	✗	✗	✓
Ease of Use via GUI	✗	✓	✓	✓	✓
Summary sentiment and score	✗	✗	✗	✗	✓

From the above analysis its clear that **GenSum** will be the first system to have the ***Domain Specific Generalization feature making this a novel feature*** compared to the other tools in the market. Furthermore, its also clear that sentiment details related to the summary isn't present in the competitor tools.

In the case of hackers stealing data from the database, **data encryption** can be applied therefore database will only contain the encrypted text data which will be then later decrypted from the decryption key when need, this will be most needed when performing the model retraining.

4.5.6 Prototyping

Table 16: Prototyping Findings (Self-Composed)

Criteria	**Discussion of findings**
To investigate the viability of proceeding with the project research, a	Throughout the iterative prototype process, the author faced various requirements and difficulties. One such challenge was the search for an appropriate dataset with the desired metadata within the movie domain.

prototype was scheduled for development.	Following a thorough assessment, the author identified a substantial dataset comprising 8 million records. However, its size and the presence of noisy text made it challenging to preprocess, although it was eventually accomplished. As the project's focus shifted towards generalization, the author explored different datasets to ensure it. The author conducted experiments using a framework known as "Optuna" to automate the hyperparameter search. Additionally, the system will undergo retraining using fresh data from the domain user, and the author intends to investigate at least three high-quality transformer designs to determine the optimal one. In preparation for the testing phase and to assess the model's generalization ability, the author gathered datasets from two domains, specifically movies and hotels. Additionally, the author explored data encryption techniques to ensure the security of the data.

4.5.6 Summary of findings

Table 17: Summary of Findings (Self-Composed)

Id	Finding	LR	Survey	Self-Evaluation	Interview	Brainstorming	Prototyping
1	The proposed system would benefit businesses (domain specific users) and general users (not domain specific)		✓			✓	
2	For the movie domain the limit of abstractive text summarization can be further pushed using optimized transformers to increase performance this being the existing limitation	✓			✓	✓	

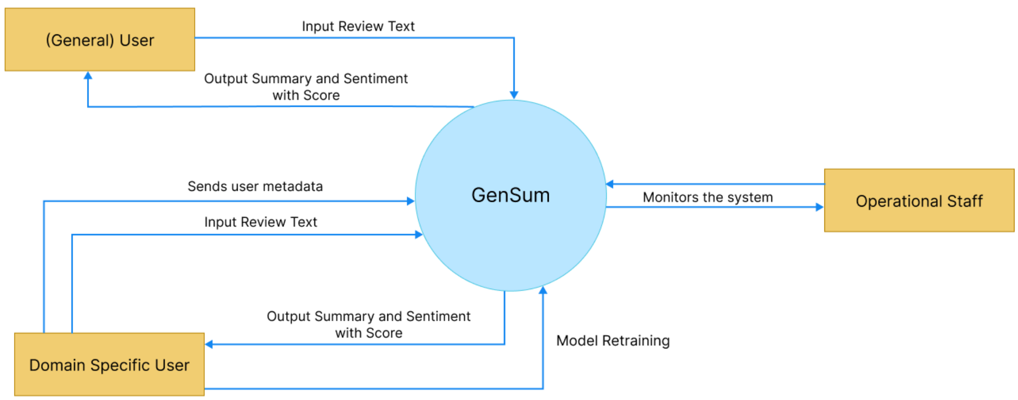
3	It's clear that customer/user reviews are valued and reviewed mostly by a vast majority of the audience before they consume or use any product or service (applies to any domain)	✓	✓		✓	✓	
4	It's clear that users spend lot of time review long reviews and they would like it being short to save time and make quicker decisions.	✓	✓			✓	
5	Hyperparameter tuning is one way to increase the performance of the system and it can be done both manually or by automated tools.	✓			✓		✓
6	Data preprocessing for the domain of Movies and Generalization is requires a lot of effort since the datasets are mostly raw data difficult to find specially in the case of movie reviews (with expected metadata)	✓					✓
7	Additional features such as sentimental and sentimental score of the review summary is mostly expected from the users.		✓				
8	Creating a hybrid transformer model to further increase the performance is a suggested improved.				✓	✓	
9	It's clear on what are the suitable evaluation metrics to be used for abstractive text summarization.	✓			✓		
10	It's clear on what the top tier transformer architecture that could be explored.	✓			✓		
11	Making use of larger new data for retraining for a specific domain, from companies/businesses who uses data which are of the same domain. (e.g.: - 50 different restaurants data can be				✓	✓	

	combined for retraining give that the domain is "Restaurants")						
12	Making use of data encryption to protect the data from hackers breaking into the database to steal data.			✓		✓	

4.6 Context Diagram

The boundaries and interactions of the system should be established before development. The graphic below shows how the system is situated.

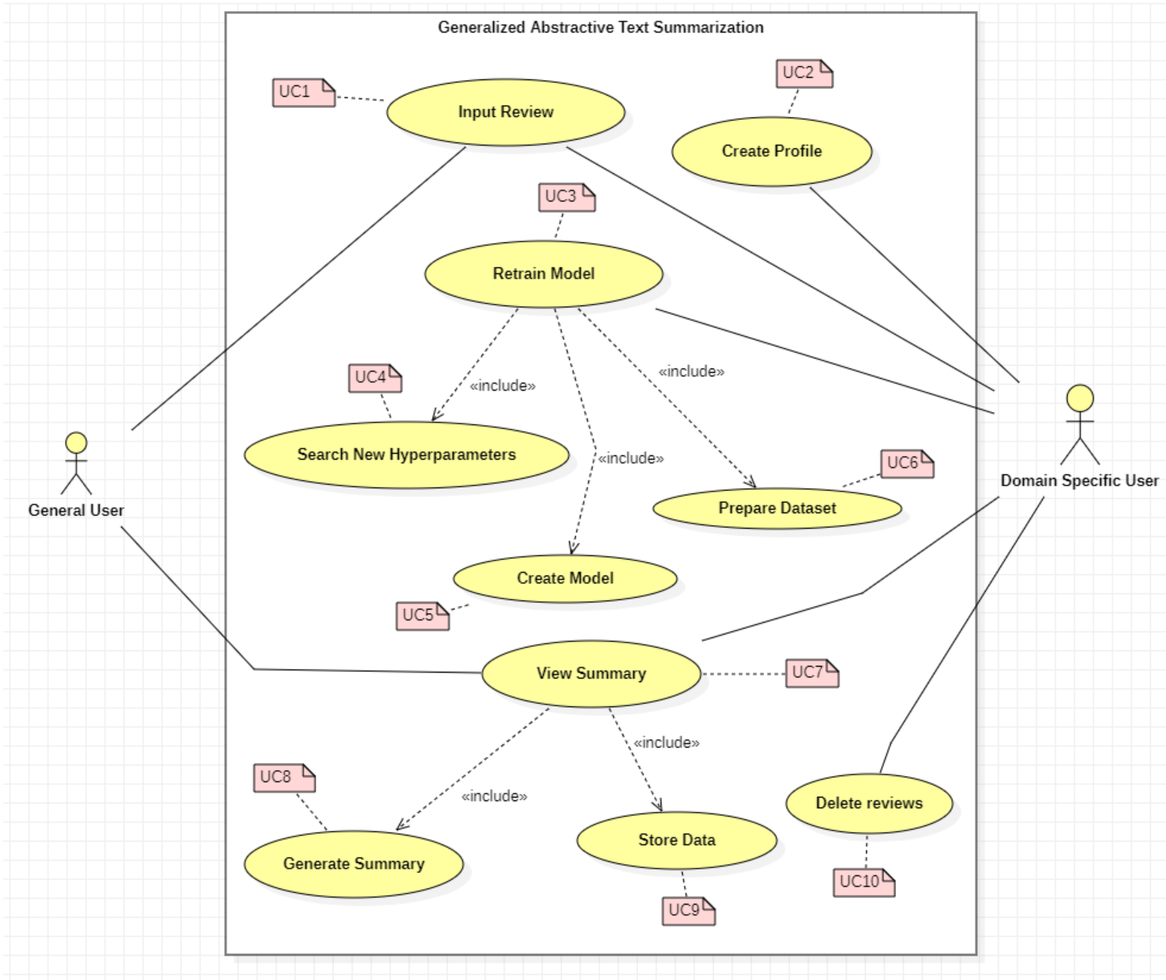
Figure 6: Context Diagram (Self-Composed)



(Area left blank on purpose, P.T.O)

4.7 Use case Diagram

Figure 7: Use Case Diagram (Self-Composed)



4.8 Use case Descriptions

Usecase diagrams with the highest importance are given below, the rest of the Usecase descriptors are available at **APPENDIX C.3.**

Table 18: Use Case Description Uc:07 (Self-Composed)

Use Case Name	View Summary
Use Case Id	UC:07

Description	Displays a summarized version of the uploaded review text from the domain user's end.	
Primary Actor	General User, Domain Specific User	
Pre-Conditions	The text review data must go through specific text preparation techniques before the summary can be produced.	
Extended use cases	None	
Included use cases	UC10, UC02	
Trigger	A user selects to summarize a given customer/user review text.	
Main flow	**Actor**	**System**
	1. The user enters the review text on the text field from the GUI. 2. Clicks on "Generate Summary" from the GUI	3. The system does the data preprocessing for the input review text. 4. Loads the generalized transformer model. 5. Generates the summary using the model. 6. **(If Domain Specific User)** stores the input review and summary into the database. 7. Returns the summary response back to the GUI
Alternative flows	None	
Expectational flows	Displays an error message if the network request fails (server is down, or internet issues from client).	
Post Conditions	Success end condition: The user is presented with the summarized review text.	

Table 19: Use Case Description Uc:03 (Self-Composed)

<table>
<tr><td>Use Case Name</td><td colspan="2">Retrain Model</td></tr>
<tr><td>Use Case Id</td><td colspan="2">UC:03</td></tr>
<tr><td>Description</td><td colspan="2">Performs model retraining with the new data from the database, to find the new best set of hyperparameters.</td></tr>
<tr><td>Primary Actor</td><td colspan="2">Domain Specific User</td></tr>
<tr><td>Pre-Conditions</td><td colspan="2">The actor should be a Domain Specific User and have an account created.</td></tr>
<tr><td>Extended use cases</td><td colspan="2">None</td></tr>
<tr><td>Included use cases</td><td colspan="2">UC05, UC06, UC07</td></tr>
<tr><td>Trigger</td><td colspan="2">The Domain Specific User clicks on the “Perform model retraining” button</td></tr>
<tr><td rowspan="2">Main flow</td><td>Actor</td><td>System</td></tr>
<tr><td>1. Domain Specific logs into their account
2. Clicks on “Perform model retraining” from the GUI</td><td>3. The system pulls all the data with respect to the user id from the database.
4. Combines data of the common domains (only if user consent is given to use their data)
5. Finds new set of hyperparameters for the model with respect to new data.
6. Trains the model using the new hyperparameters.
7. Saves the model with the user Id</td></tr>
</table>

		8. Updates the status in the database if succeed/fails
Alternative flows	None	
Expectational flows	Displays an error message if the network request fails (server is down, or internet issues from client).	
Post Conditions	Success end condition: The user will be able to see the recent status of the model if the retraining is successful or failed	

4.9 Requirements

4.9.1 Functional Requirements

Based on the significance of the system demands, the 'MoSCoW' approach was utilized to identify their priority levels. The details related to the priority levels are detailed at **APPENDIX C.4.**

The Usecase description along its mapping id is also listed at **APPENDIX C.3.**

Table 20: Functional Requirements (Self-Composed)

FR ID	**Requirement**	**Priority Level**	**Use Case**
FR1	Both general and domain specific users must be able to enter a review text from the GUI considering as the starting point of the summary generation.	M	UC01
FR2	Only Domain Specific Users should be able to sign up and create an account after entering the necessary details required	S	UC02
FR3	The system could allow the ability to update the account details of the domain user after creating the account	C	UC02
FR4	The system must undergo model retraining with the new data stored in the database for the specific domain user, when its triggered from the GUI with the user's consent.	M	UC03

FR5	The system could be able to perform model retraining automatically during off peak hours every day.	C	UC03
FR6	The system must be able to find the new set of best hyperparameters with the usage of the new data.	M	UC04
FR7	The system must be able to able to retrain the model with the new best hyperparameters and create the model	M	UC05
FR8	The system must be able to pull the new data from the database to recreate the new dataset for retraining.	M	UC06
FR9	The system should be able to combine all the data from a common group of domains when creating the dataset only given that the consent is approved to use their data	C	UC06
FR10	The system must be able to process the review text and display the summary output on the GUI	M	UC07
FR11	The system must be able to use the latest trained model to generate the summary for the review text	M	UC08
FR12	The system could also find the sentiment of the generated summary if its positive or negative and return the result.	C	UC08
FR13	The system could make use of a hybrid model for the text summarization.	C	UC08
FR14	The system must store the entered user review and generated summary to be stored in the database for retraining purposes.	M	UC09
FR15	The system should encrypt the data when saving into the database (both the review and summary)	S	UC09
FR14	The system could allow the domain users to delete the reviews from the database.	C	UC10

(P.T.O)

4.9.2 Non-Functional Requirements

The non-functional requirements are prioritized into two level of which are “Important” and “Desirable”

Table 21: Non-Functional Requirements (Self-Composed)

NFR ID	Requirement	Specification	Priority Level
NFR1	The system needs to be simple enough for non-technical individuals to utilize without much effort.	Usability	Important
NFR2	Meaningful error messages should be displayed if anything goes wrong	Usability	Desirable
NFR3	Summary generation should be done within 3000ms	Performance	Important
NFR4	Following coding standards and best practices	Maintainability	Important
NFR5	Any domain users are able to use the application and model performance will adapt with respect to the domain	Generalization	Important
NFR6	The system should protect against data corruption by attackers, and testing can ensure this.	Security	Desirable
NFR7	The prototype can be used by several domains and multiple businesses under a single domain, then the system may have to support many concurrent user-requests.	Scalability	Desirable

4.10 Chapter summary

In this chapter, a Rich Picture Diagram was created to show how the system interacts with society and stakeholders. The Saunder's Onion model was used to depict stakeholders and requirement gathering techniques were used to collect data and opinions. The use cases, functional and non-functional requirements were specified using the information gathered from the requirement elicitation methodologies.

CHAPTER 05. SOCIAL, LEGAL, ETHICAL & PROFESSIONAL ISSUES

5.1 Chapter overview

In this chapter, the social, legal, ethical, and professional concerns that emerged during the course of this project are delineated, along with the measures that were implemented to address them.

5.2 SLEP issues and mitigation

The table presented below provides a detailed analysis of the identified social, legal, ethical, and professional (SLEP) issues, as well as the corresponding mitigation strategies that were employed.

Table 22: SLEP Issues & Mitigation

Social	Legal
♦ The dissertation refrains from mentioning the names of the interviewees. ♦ During the conducted survey, no personal details were collected from the respondents to ensure their anonymity. The respondents were kept anonymous throughout the survey process. ♦ The information obtained during the research was not saved or stored elsewhere, ensuring that the data remained secure and confidential.	♦ All the datasets used in the research are freely available for the public to access. ♦ The tools and technologies utilized in the research are either open source or provided to students free of charge, thereby enabling wider access and use by researchers and students. ♦ The codebase utilized in the research is available on GitHub under the open-source MIT license, allowing others to access and use it freely.
Ethical	**Professional**
♦ The participants were provided with clear information regarding the intended use of the data that was collected from them. ♦ The work presented in this report is entirely original and has not been reproduced	♦ Throughout the project, the team made every effort to adhere to professional standards to ensure the quality and integrity of the work.

elsewhere. All the ideas taken from external sources have been appropriately cited, and credit has been given where it is due.	♦ The responses and evaluations received during the research were not tampered with, and any limitations encountered during the study were clearly stated and acknowledged.

5.3 Chapter summary

In this chapter, the author has expounded upon the social, legal, ethical, and professional issues that were encountered during the research process, and has outlined the measures taken to address and resolve them.

CHAPTER 06. DESIGN

6.1 Chapter overview

This chapter discusses the design decisions made to establish an appropriate architecture for implementation based on the requirements obtained. High-level design, low-level design, design diagrams, and UI wireframes have been used to describe how the design goals are meant to be achieved while demonstrating the reasoning for selected design decisions.

6.2 Design goals

The design goals that should be achieved by the system are specified in the table below.

Table 23: Design Goals of The Proposed System (Self-Composed)

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.
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.

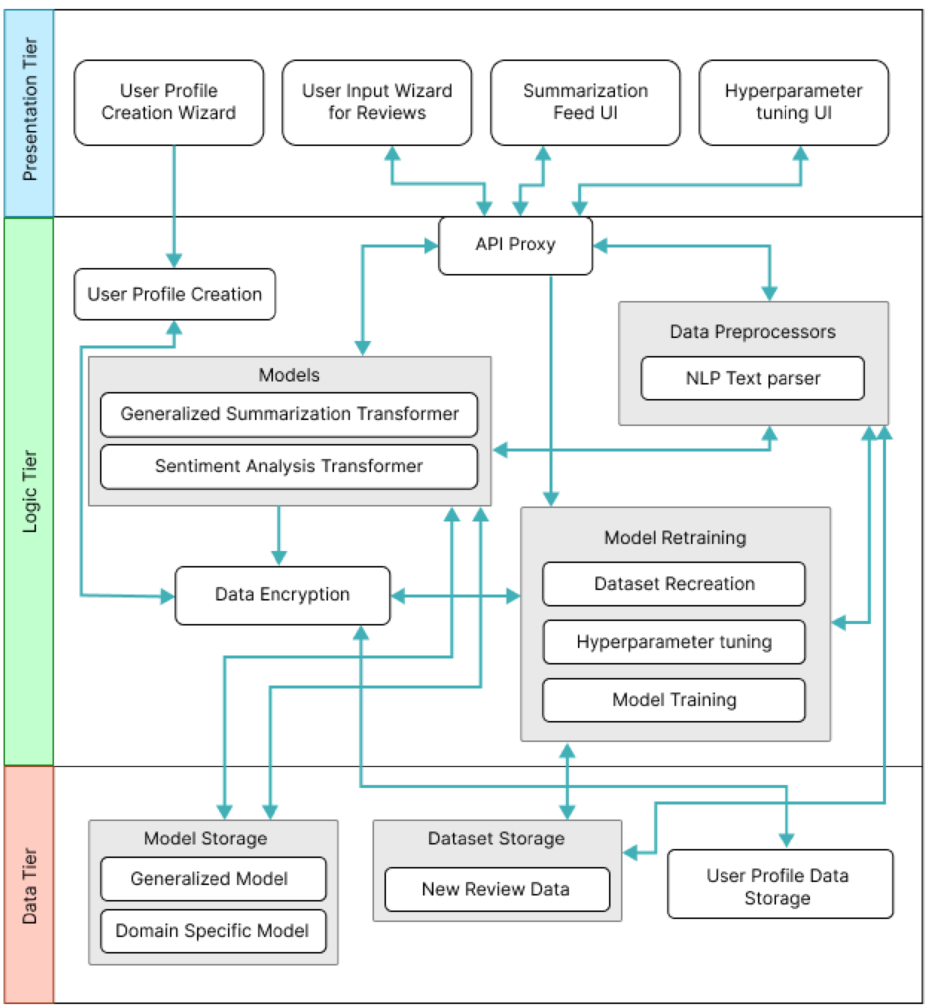
Scalability	In a production environment, the system may need to accommodate a large number of concurrent user requests. This should be manageable by the backend. The system should be easily expandable to accommodate new data.

6.3 High Level Design

6.3.1 Architecture Diagram

The system has a three-tier architecture that separates the data, logic, and presentation levels. The research contribution of the system includes generalization and domain-specific adaptive hyperparameter tuning and data preprocessing.

Figure 8: Three-Tiered Architecture (Self-Composed)



6.3.2 Discussion of tiers of the architecture

Data Tier

1. **Model Storage** - The text summarization models which will be used for both generalized text summarization and domain specific text summarization will be stored here.
   a) **Generalized Model** – The model which will be used by general users to generated review summarized, this model will be hyperparameter tuned for genialized purpose.
   b) **Domain Specific Model** – The model will be used by domain specific users for review summarization, this model will be replaced whenever the model retraining is triggered from the domain user.
2. **Dataset Storage** – The data which is required for model training will be available.
   a) **New Review Data** – The data stored here are from the domain users when they use the application, the data will be storage and used for retraining.
3. **User Profile Data Storage** – The metadata data related to the domain specific user when creating the user profile will be stored, for updating and profile deletion.

Logic Tier

1. **User Profile Creation** – Allowing to create unique user profiles for each domain user, main purpose comes when working with model retraining to figure out the data to be used.
2. **API Proxy** – Interface which allows the frontend to communicate with the backend services via HTTP calls/ request.
3. **Data Preprocessors** – The text data that will be used as input for the text summarizer must be cleaned using the preprocessing code.
   a) **NLP Text parser** – Responsible for cleaning the input text review when received from the API endpoint.
4. **Models** – The model which will be responsible in generating the summary from the input review and find the sentiment of the summary generated.
   a) **Generalized Summarization Transformer** – This is the summarization model which will be used, an adaptive model depending on the domain and type of user interacting with the model with optimized hyperparameters.
   b) **Sentiment Analysis Transformer** – This model will be used to classify the generated summary into positive or negative sentiment.

5. **Data Encryption** – Data encryption is in charge of data protection/safety, keeping domain data extremely secure and leaving it useless even if it is stolen.
6. **Model Retraining** – Responsible for retraining the model with new data and finding new set of hyperparameters.
    a) **Dataset Recreation** – Responsible for recreating the dataset with new data which has been given as input from the domain users
    b) **Hyperparameter tuning** – Responsible for finding the new best set of hyperparameters using the new data.
    c) **Model Training** – Responsible for training the new model with the new set of hyperparameters found.

Presentation Tier

1. **User Profile Creation Wizard** – The UI that presents the user to create a new profile if they are planning to use the software for their domain business, or a general user to skip the sign up if only a generalized summary is required.
2. **User Input Wizard for Reviews** – The UI that will request the user to input the review which needs to be summarized.
3. **Summarization Feed UI** – The UI that displayed the summarized text for the input review.
4. **Hyperparameter tuning UI** – The UI that triggers model retraining when the domain user performs an action on it.

6.4 System Design

6.4.1 Choice of Design Paradigm

The main reason behind the author going ahead with **SSADM (Structured Systems Analysis and Design Method)** over **OOAD (Object-Oriented Analysis and Design)** to build the protype was due to the ease of ability to extend the system features when it comes to future developments of the system. Given below are the other factors as to why the choice of SSADM was considered:

- Object Oriented approaches will not make a greater benefit since the main core project research lies towards Data Science.
- Ability to demonstrate the MVP (Minimum Viable Product) prototype implementation for the research application is more convenient.

- More time efficient when concerned with the time constraint of having to complete the documentation research along with the project implementation.

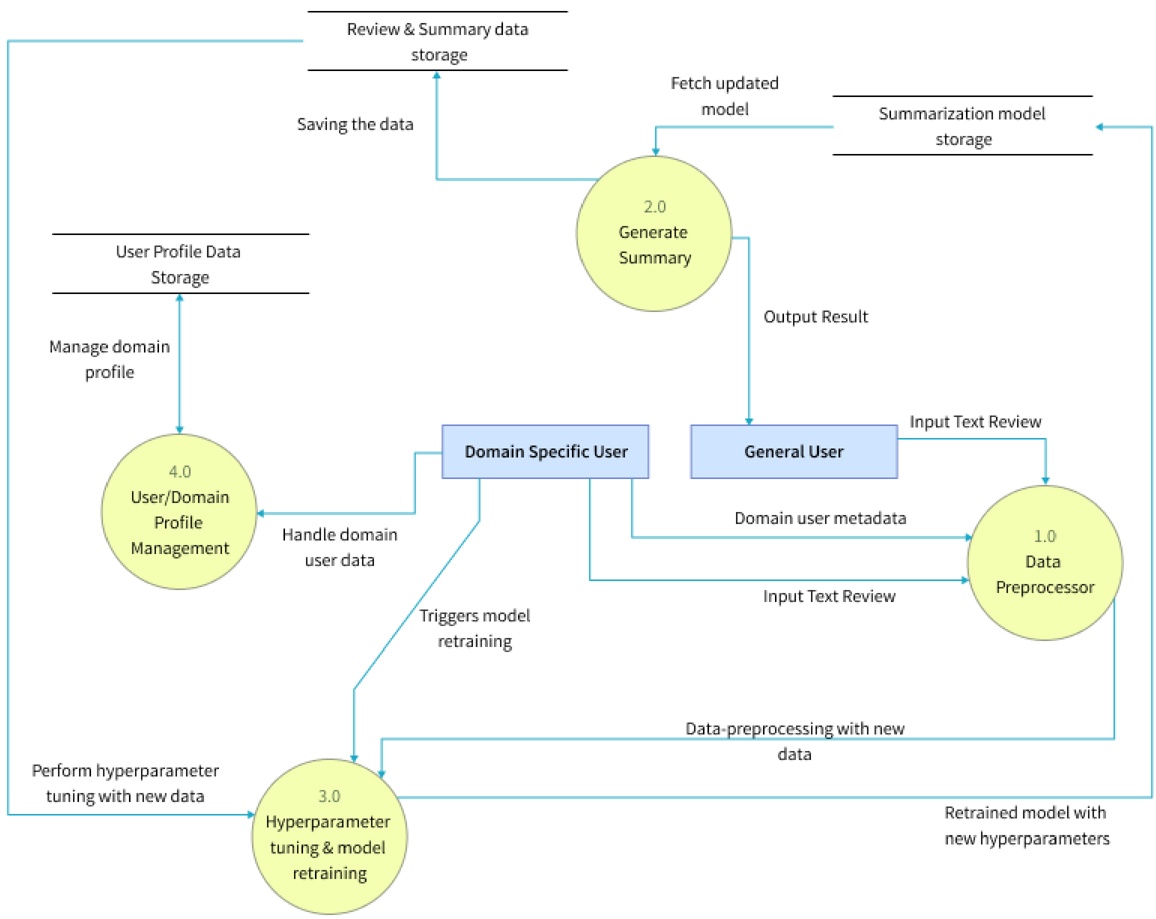
6.5 Design diagrams

6.5.1 Data Flow diagrams

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.

6.5.1.1 Level 01 Data Flow diagram

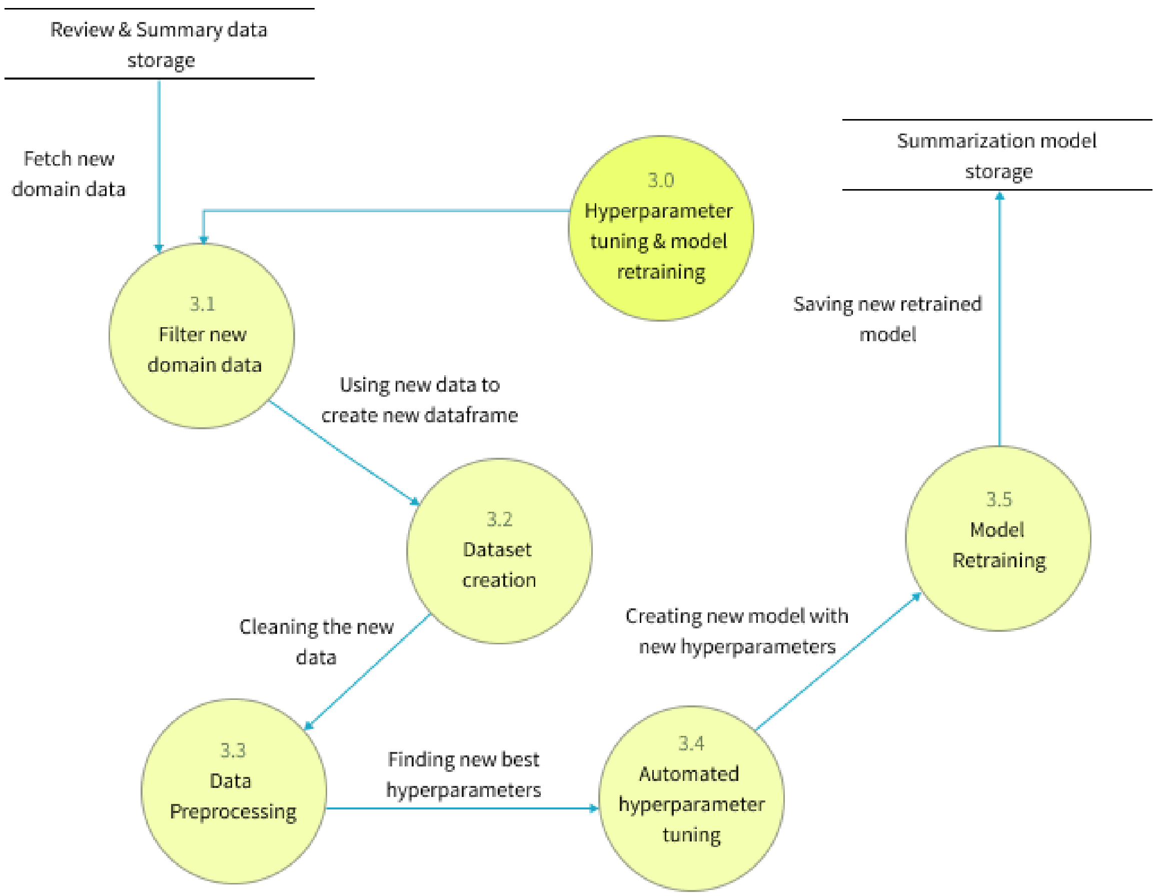
Figure 9: Data Flow Diagram - Level 01 (Self-Composed)



6.5.1.2 Level 02 Data Flow diagram

The level 02 data flow diagram given below is a further breakdown of the core hyperparameter tuning and model retraining proposed in the level 01 data flow diagram.

Figure 10: Data Flow Diagram - Level 02 (Self-Composed)



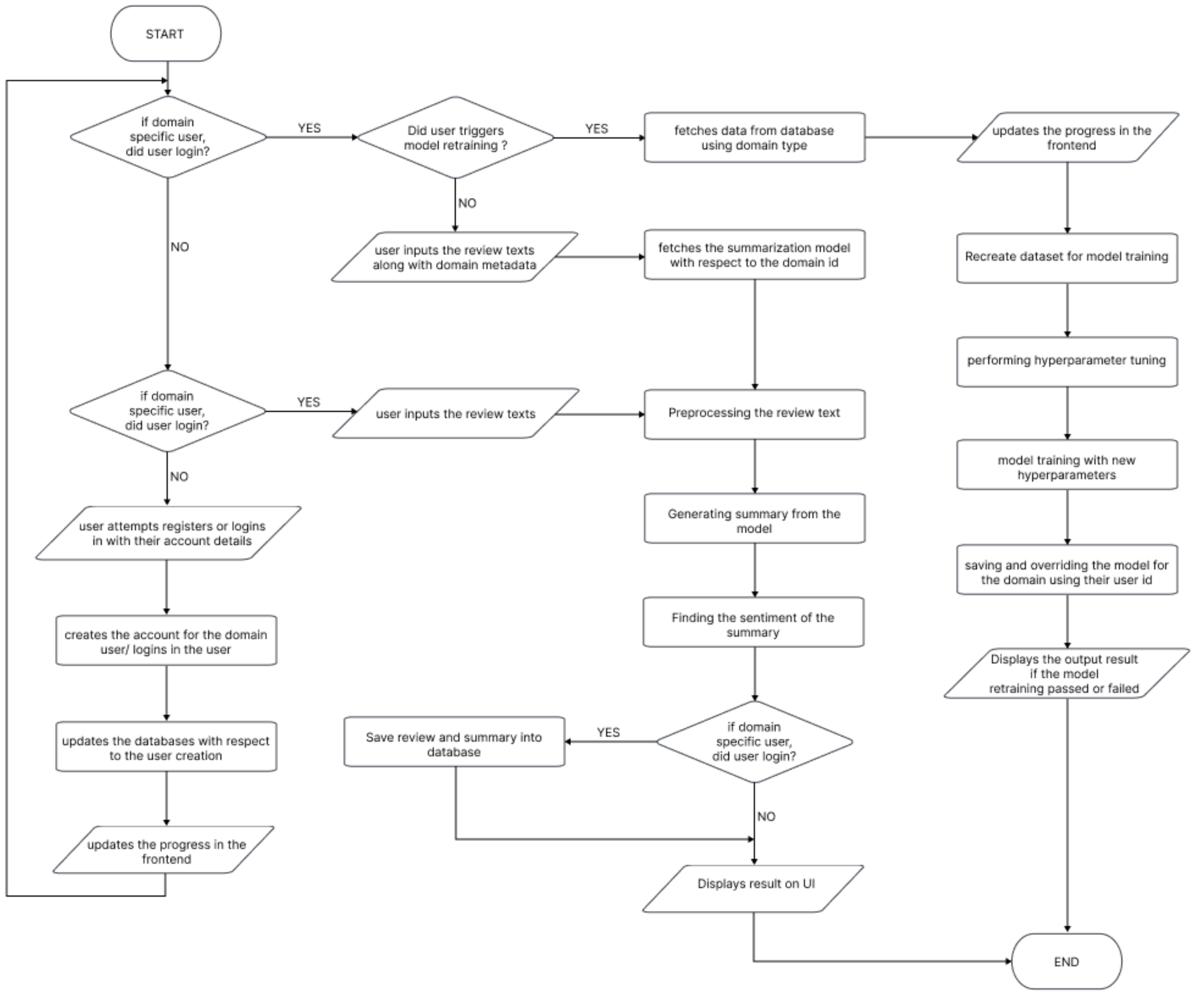
6.5.4 UI design

Given the specifications acquired from the target audience, the author chose a web application for the simulation of the proof-of-concept application. A wireframe design was created to depict the key user interface aspects in the system and is available in **APPENDIX D.1.**

6.5.5 System process activity diagram

The flowchart given below represents the algorithm's flow and the decision structures which explains the flow of the system which is initially expected requirement.

Figure 11: System Process Flow Chart (Self-Composed) – *(view high qual version)*



6.6 Chapter summary

This chapter provides an in-depth examination of the project's design, including its architectural features and explains the core flow via data flow diagrams. The chapter concludes with a preview of the user interface wireframes that will be utilized to facilitate interaction between the end-user and the system.

CHAPTER 07. IMPLEMENTATION

7.1 Chapter overview

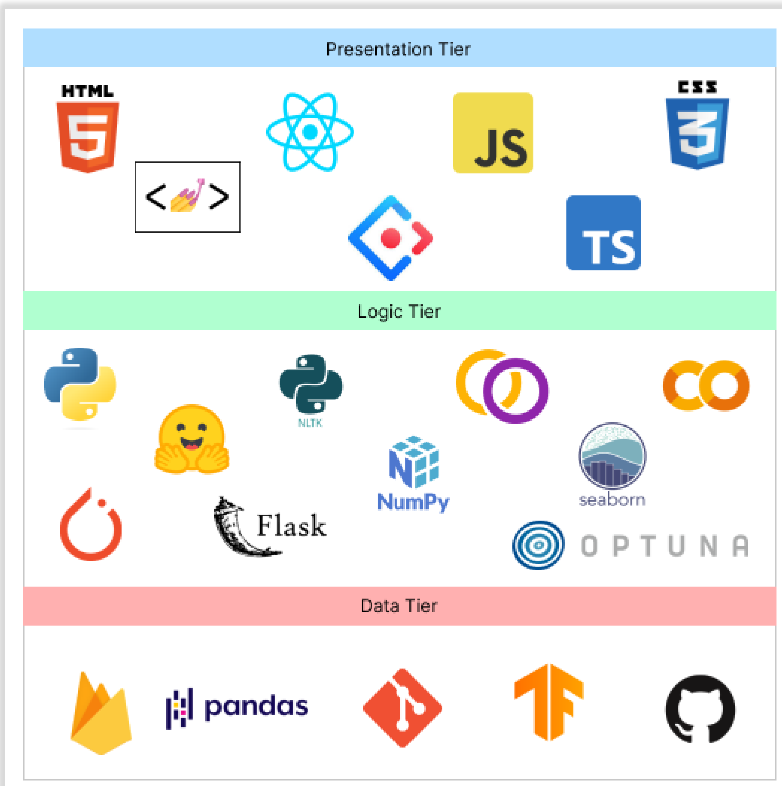
This chapter will provide a thorough overview of the technologies, supporting tools, and languages utilized for the project development, as well as the fundamental implementation of the research prototype.

7.2 Technology selection

7.2.1 Technology stack

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

Figure 12: 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.

7.2.2 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.

The table below shows the datasets which have been used by previous researchers, therefore this can experiment for the protype development.

Table 24: Dataset Sources (Self-Composed)

Dataset	**Source**	**Purpose**
CNN Dailymail	TensorFlow Datasets	Model Generalization
Gigaword	TensorFlow Datasets	Model Generalization
Xsum	TensorFlow Datasets	Model Generalization
Amazon movie reviews	Stanford University Data	Domain Testing
Hotel Reviews	Kaggle	Domain Testing

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, <u>Xsum</u> performed the best, so it was selected as the final dataset for the project. All resources related to the dataset used can be found in **APPENDIX E.1.**

7.2.3 Programming Language Selection

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.

TypeScript (it's a superset of JavsScript) was chosen for the frontend development in order to display dynamic content and create a highly interactive and engaging user experience. A further analysis of this can be found at **APPENDIX E.2**

7.2.4 Development Framework Selection

The author has chosen several development frameworks for the project covering all areas, the table given below describes the purpose of choosing each framework and whats it used for in the project.

The author has conducted an analysis on the deep learning frameworks at **APPENDIX E.3,** an analysis on the UI frameworks at **APPENDIX E.4** and about the API frameworks at **APPENDIX E.5**

Table 25: Development Framework Utilized (Self-Composed)

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 library 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.

7.2.5 Libraries Utilized

Table 26: Libraries Used with Reasonings (Self-Composed)

Library	**Reasoning for selection**
Firebase	Used for providing backend services for mobile and web application development.
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.
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.

7.2.6 IDE's Utilized

Table 27: Ide's Used Along with Justifications (Self-Composed)

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.

7.2.7 Summary of Technology Selection

Table 28: Summary of Technology Selection (Self-Composed)

Component	Tools
Programming Languages	TypeScript, Python
Development Framework	Flask, PyTorch, Optuna
UI Framework	Ant Design, React
Libraries	NLTK, Rouge, React, Pandas, Gramformer, Matplotlib & Seaborn, Axios, Transformers (from hugging face)
IDE – Research	Google Colab, Jupyter Notebook
IDE – Product	VSCode
Version Control	Git, GitHub
Data storage	Firebase

7.3 Implementation of Core Functionalities

The project's core functionalities include the experiments of top-tier transformer architectures to determine the optimal one, applying data preprocessing steps, automating hyperparameter searching, retraining the model with new data fetched from the database and new hyperparameters, and having the model be able to summarize reviews from both domain users and general users.

7.3.1 Automated Hyperparameter Search & Model Training

The author did a research on different approaches to automate the hyperparameter searching, because manual hyperparameter tuning is total waste of time. Multiple hyperparameter tuning frameworks were available, however Optuna was chosen due to its flexibility and ease of use.

Figure 13: Hyperparameter Range Initialization (Self-Composed)

```
# Specify our parameter range and project variables
LR_MIN = 4e-5
LR_CEIL = 0.01
WD_MIN = 4e-5
WD_CEIL = 0.01
MIN_EPOCHS = 8
MAX_EPOCHS = 15
PER_DEVICE_EVAL_BATCH = 4
PER_DEVICE_TRAIN_BATCH = 4
MIN_BATCH_SIZE = 4
MAX_BATCH_SIZE = 6
NUM_TRIALS = 1
SAVE_DIR = 'opt-test'
MODEL_NAME = 'facebook/bart-base'
MAX_INPUT = 512
MAX_TARGET = 128
```

The given code initializes hyperparameters with a group of values, some of which may be within a specified range based on the min and max parameters. These parameters will be used during hyperparameter search training to determine the optimal values. If no range is specified, the default will start at zero.

Figure 14: Hyperparameter Search Using Optuna (Self-Composed)

```
print_custom('Performing hyperparameter training....')
def objective(trial: optuna.Trial):
    # Specify the training arguments and hyperparameter tune every arguments which are possible to tune
    training_args = Seq2SeqTrainingArguments(
        output_dir=SAVE_DIR,
        save_strategy="epoch",
        evaluation_strategy="epoch",
        learning_rate=trial.suggest_float("learning_rate", LR_MIN, LR_CEIL, log=True),
        weight_decay=trial.suggest_float("weight_decay", WD_MIN, WD_CEIL, log=True),
        num_train_epochs=trial.suggest_int("num_train_epochs", MIN_EPOCHS, MAX_EPOCHS),
        warmup_ratio=trial.suggest_float("warmup_ratio", 0.0, 1.0),
        per_device_train_batch_size=trial.suggest_int("per_device_train_batch_size", MIN_BATCH_SIZE, MAX_BATCH_SIZE),
        per_device_eval_batch_size=trial.suggest_int("per_device_eval_batch_size", MIN_BATCH_SIZE, MAX_BATCH_SIZE),
        save_total_limit=1,
        load_best_model_at_end=True,
        greater_is_better=True,
        predict_with_generate=True,
        run_name=MODEL_NAME,
        report_to="none",
    )

    # Create the trainer
    trainer = Seq2SeqTrainer(
        model=model,
        args=training_args,
        data_collator=data_collator,
        train_dataset=tokenize_data["train"],
        eval_dataset=tokenize_data["test"],
        tokenizer=tokenizer,
    )

    # Train the model
    trainer.train()

    # Evaluate the model
    metrics = trainer.evaluate()

    torch.cuda.empty_cache()

    # Return the loss
    return metrics["eval_loss"]
```

The above snippet shows how the Optuna framework is integrated with the model training code to perform automated hyperparameter search. The main performance contributing parameters are considered for the hyperparameter searching this includes learning rate, weight decay, num of training epochs, warmup ratio, batch size.

Figure 15: Hyperparameter Results and Training Arguments (Self-Composed)

```
# Hyperparameter results
learning_rate = study.best_params['learning_rate']
weight_decay = study.best_params['weight_decay']
num_train_epochs = study.best_params['num_train_epochs']
warmup_ratio = study.best_params['warmup_ratio']
per_device_train_batch_size = study.best_params['per_device_train_batch_size']
per_device_eval_batch_size = study.best_params['per_device_eval_batch_size']
```

```
args = transformers.Seq2SeqTrainingArguments(
    'generalization-summary',
    learning_rate=learning_rate,
    weight_decay=weight_decay,
    warmup_ratio=warmup_ratio,
    num_train_epochs=num_train_epochs,
    per_device_train_batch_size=per_device_train_batch_size,
    per_device_eval_batch_size= per_device_eval_batch_size,
    save_total_limit=2,
    eval_accumulation_steps=1,
    predict_with_generate=True,
    evaluation_strategy='epoch',
    gradient_accumulation_steps=2,
    fp16=True
  )
```

The above snippet demonstrates how to result of the hyperparameter search is used within the training arguments for model training.

Figure 16: Model Training (Self-Composed)

```
trainer = transformers.Seq2SeqTrainer(
    model,
    args,
    train_dataset=tokenize_data['train'],
    eval_dataset=tokenize_data['validation'],
    data_collator=data_collator,
    tokenizer=tokenizer,
    compute_metrics=compute_rouge
)
trainer.train()
```

The above code snippet is the model training initiation with the optimal hyperparameters.

7.3.2 Model Usage General & Domain Specific Users

Figure 17: General User Review Text Summarization (Self-Composed)

```
@app.route('/text-summarizer/general', methods=['POST'])
def getGeneralizedSummary():
    try:
        data = request.get_json()
        review = data['review']
        inputs = generalized_tokenizer.encode(review, return_tensors='pt', max_length=MAX_INPUT, truncation=True)
        outputs = generalized_model.generate(inputs, max_length=150, min_length=40, length_penalty=2.0, num_beams=4,
        early_stopping=True)
        summary = generalized_tokenizer.decode(outputs[0], skip_special_tokens=True)

        sentimentAnalysisOutput = query({ "inputs": summary })
        sentiment, score = getOverallSentimentWithScore(sentimentAnalysisOutput)
        return {'summary': summary, 'sentment': {
            'sentiment': sentiment,
            'score': score
        } }, 200
    except Exception as e:
        return {'message': str(e)}, 500
```

The above code snippet is an API endpoint which handles text (review) summarization for the general users where they don't need to create and account or have specialized model assigned to them instead the general model is utilized.

Figure 18: Assigning A Specific Model for The New Domain User (Self-Composed)

```
@app.route('/domain-profile-creation', methods=['POST'])
def createDomainUserProfile():
    try:
        data = request.get_json()
        userId = data['userId']

        folder_path = 'model/' + userId
        model_path =  folder_path + '/' + MODEL_NAME
        tokenizer_path = folder_path + '/' + TOKENIZER_NAME

        if not os.path.exists(folder_path):
            os.mkdir(folder_path)

        generalized_model.save_pretrained(model_path)
        generalized_tokenizer.save_pretrained(tokenizer_path)

        return {'message': "Successfully created the model"}, 200
    except Exception as e:
        return {'message': str(e)}, 500
```

The above code snippet describes an API for assigning a copy of the generalized model for the user id of the domain (given that the domain user signed up for the application), the reason for creating a copy is for retraining purposes with new data.

Figure 19: Domain Specific Text Review Summarization (Self-Composed)

```
@app.route('/text-summarizer/domain', methods=['POST'])
def getDomainSpecificSummary():
    try:
        data = request.get_json()
        review = data['review']
        userId = data['userId']

        folder_path = 'model/' + userId
        model_path =  folder_path + '/' + MODEL_NAME
        tokenizer_path = folder_path + '/' + TOKENIZER_NAME

        if not os.path.exists(folder_path):
            return {'message': "Model not found"}, 404

        model = transformers.AutoModelForSeq2SeqLM.from_pretrained(model_path)
        tokenizer = transformers.AutoTokenizer.from_pretrained(tokenizer_path)

        inputs = tokenizer.encode(review, return_tensors='pt', max_length=MAX_INPUT, truncation=True)
        outputs = model.generate(inputs, max_length=150, min_length=40, length_penalty=2.0, num_beams=4, early_stopping=True)
        summary = tokenizer.decode(outputs[0], skip_special_tokens=True)

        sentimentAnalysisOutput = query({ "inputs": summary })
        sentiment, score = getOverallSentimentWithScore(sentimentAnalysisOutput)

        db.collection('domainUsers').document(userId).collection('reviewData').add({
            'review': review,
            'summary': summary,
            'sentiment': sentiment,
            'score': score
        })

        return {'summary': summary, 'sentment': {
            'sentiment': sentiment,
            'score': score
        } }, 200
    except Exception as e:
        return {'message': str(e)}, 500
```

The code uses a domain-specific model to generate a summary, saves the input/output to a database, and analyzes sentiment using a pre-trained transformer from Hugging Face API.

7.3.3 Model Retraining

Figure 20: Fetching Related Data for Model Retraining (Self-Composed)

```
@app.route('/domain-profile-retraining', methods=['POST'])
def retrainDomainSpecifcModel():
    try:
        data = request.get_json()
        newReviewSummaryData = []

        userId = data['userId'] # The user id is only needed to save the model in the respective folder
        domainType = data['domainType'] # Using the domainType, we can get all the data from other users which have been given
        access for retraining
        isUseOtherData = data['isUseOtherData'] # we can have a radio button in the frontend to select if the user wants to
        retrain only with their data or with the other users data as well

        # Steps to be considered for retraining the model and dataset recreation
        # 1. By checking the isAccessible flag, we can decide whether to use the data for model retraining, then we get all the
        data from the database which isAccessible = true for the given domainType
        print('Fetching data from the database...')
        if isUseOtherData == True:
            users = db.collection('domainUsers').where('domainType', '==', domainType).where('isAccessible', '==', True).get()
            for user in users:
                reviewData = db.collection('domainUsers').document(user.id).collection('reviewData').get()
                for review in reviewData:
                    newReviewSummaryData.append(review.to_dict())

        else:
            user = db.collection('domainUsers').document(userId).get()
            if user.exists:
                reviewData = db.collection('domainUsers').document(userId).collection('reviewData').get()
                for review in reviewData:
                    newReviewSummaryData.append(review.to_dict())
            else:
                return {'message': "User not found"}, 404
        print('Successfully fetched data from the database')
```

The code snippet above describes the necessary data fetched from the database to create the new dataset for model retraining, once the new dataset is created it is passed through a function to perform hyperparameter tuning and then retrain the model. Once completed retraining, the old model will be replaced with the new model in the folder path location.

7.3.4 Data Preprocessing

The raw dataset was contaminated with a lot of noise, numerous data preprocessing steps were required to clean the data before model training. The related preprocessing scripts can be found at **APPENDIX E.6**.

7.4 User interface

Screenshots of the final GUI are placed under **APPENDIX E.7**.

7.5 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, moreover the testing and evaluation related code for the models is discussed.

CHAPTER 08. TESTING

8.1 Chapter Overview

After achieving an acceptable level of implementation, it is imperative to subject the system to rigorous testing to ascertain that its functionalities operate as intended. This chapter entails conducting comprehensive testing on both the system and the utilized model. The testing methodologies employed encompass functional, non-functional, integration, and model testing, all aimed at providing an extensive evaluation of the system's performance.

8.2 Testing objectives & goals

The primary objective of testing is to verify that the system functions in the expected manner. Achieving this objective requires meeting several testing goals.:

- Confirm the model's performance is optimized.
- Ensure that the implemented functionalities are in line with the "Must have" and "Should have" criteria of the MoSCoW technique.
- Identify any necessary bug fixes or improvements that need to be applied to the application.
- Ascertain if the critical non-functional requirements have been satisfied.
- Perform baseline benchmarking to establish a standard for comparing the system's future performance.

8.3 Testing criteria

Before proceeding with the testing phase, a specific set of standards was established to assess the system using two different methods.

- **Functional Quality** – This involves examining the system's developmental traits and technical specifications to determine its conformity to the designated design based on functional requirements.

- **Structural Quality** – This evaluates the system's non-functional requirements while simultaneously ensuring that it satisfies the functional requirements' performance criteria.

8.4 Model testing & evaluation

8.4.1 Model testing

Three transformer architectures, considered to be the most prominent, were employed to train the datasets, and subsequently, conducted testing on all of them. The figures presented below display the validation accuracy and loss for each of the three models, facilitating the selection of the model that performed the best.

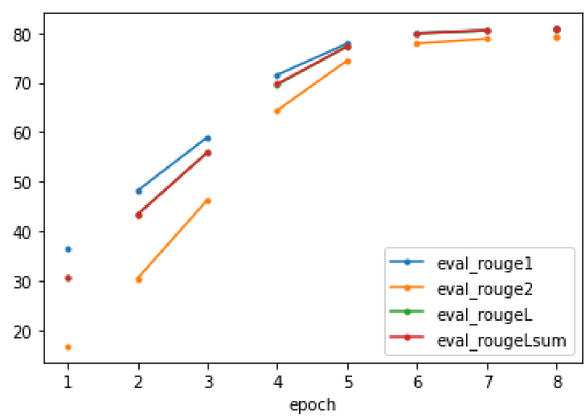


Figure 21: Validation Accuracy by number of epochs – bart model (*Self-Composed*)

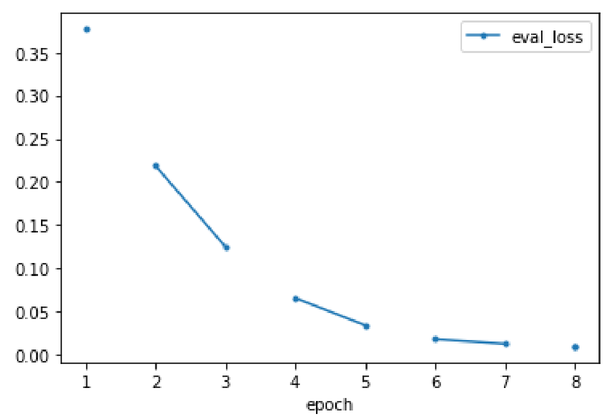


Figure 22: Validation Loss by number of epochs – bart model (*Self-Composed*)

The figures shown above depict the outcomes of testing the BART model, where all the rouge scores exhibit an upward trend as the number of epochs increases.

These results were obtained after conducting hyperparameter tuning, which yielded a higher score compared to the benchmark scores reported in earlier research.

Similarly, there is a considerable decline in the validation loss as the number of epochs increases, indicating that the model is enhancing its capability to generate precise predictions.

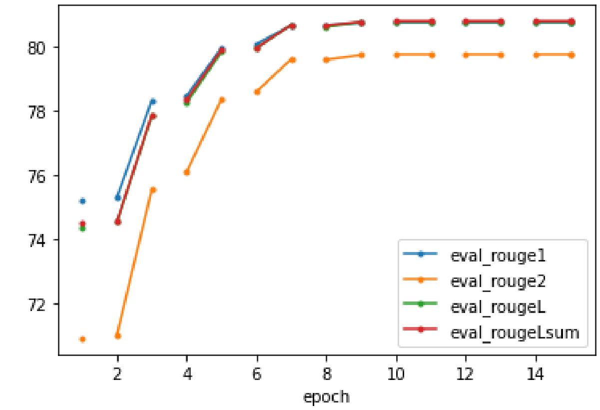


Figure 23: Validation Accuracy by number of epochs – t5 model (*Self-Composed*)

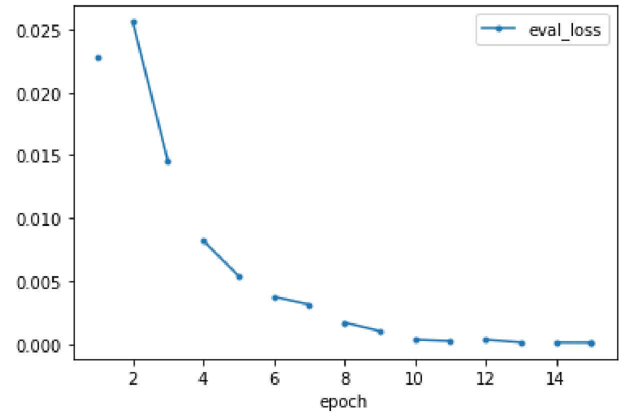


Figure 24: Validation Loss by number of epochs – t5 model (*Self-Composed*)

The presented figures demonstrate the testing results of the T5 model, where all the rouge scores exhibit an exponential increase and eventually plateau as the number of epochs increases. Similarly, the validation loss also shows an exponential decrease with respect to the number of epochs. Nevertheless, the overall results indicate that the BART model performs slightly better than the T5 model.

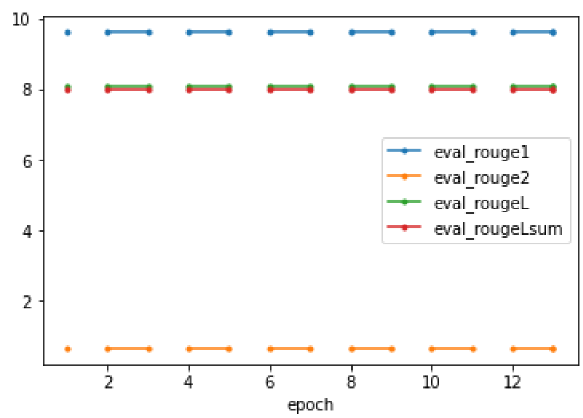


Figure 25: Validation Accuracy by number of epochs – Pegasus model (*Self-Composed*)

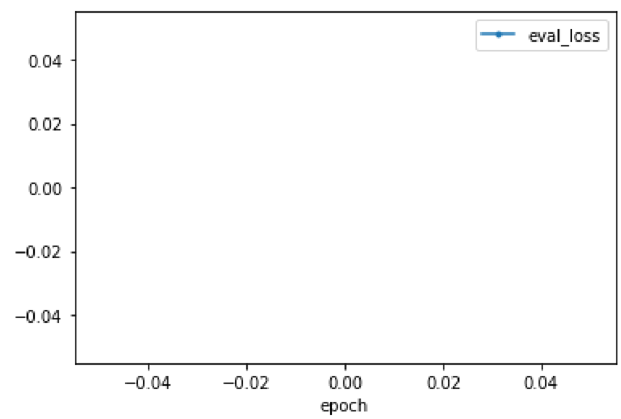


Figure 26: Validation Loss by number of epochs – Pegasus model (*Self-Composed*)

The illustration above depicts the testing outcomes of the Pegasus model, indicating a poor performance despite identifying the optimal hyperparameters for the given training dataset. The rouge scores remain consistently low, and the validation loss graph depicts no variation due to the lack of improvement in accuracy.

8.4.2 Model Evaluation

The evaluation metrics for the models were computed based on the recommended criteria from the literature review and the author's proficiency in evaluating machine learning models. These metrics were then presented in **LR Chapter**, under the "**Evaluation**" topic.

According to previous research conducted by (Steinberger and Jezek, 2009), among the primary scoring methods of ROUGE and BLEU, ROUGE and its metric versions are regarded as the most appropriate for achieving optimal results.

Table 29: Model Evaluation Results

Model	Rouge1	Rouge2	RougeL	RougeLSum
Bart	80.78	79.42	80.80	80.83
T5	80.75	79.76	80.77	80.79
Pegasus	10.07	1.21	8.34	8.33

According to the evaluation provided, it is evident that the Bart transformer model performs better than the other two models, while Pegasus has the poorest performance.

The inadequate performance of Pegasus model could be due to multiple reasons even if it has undergone optimization and suitable preprocessing. Pegasus and Bart have different model architectures which can impact how they comprehend and represent text data. Pegasus uses a transformer-based encoder-decoder framework while Bart uses a denoising autoencoder framework. It's possible that one architecture may be better suited for a specific dataset than the other.

Domain Model Evaluation

The adaptive generalized solution was tested with two different domains - **movies** and **hotels** - to demonstrate the effectiveness of the approach.

Testing dataset:

1. **Movie Review**: The dataset was provided by Stanford University, which are amazon movie review data (SNAP, no date).
2. **Hotel Review**: The dataset was provided by Kaggle (Jonathanoheix, 2018).

The following are the evaluation results for each domain when tested with the corresponding dataset listed above after model retraining with unseen data.

Figure 27: Hotel Domain (Cinnamon Type) Evaluation

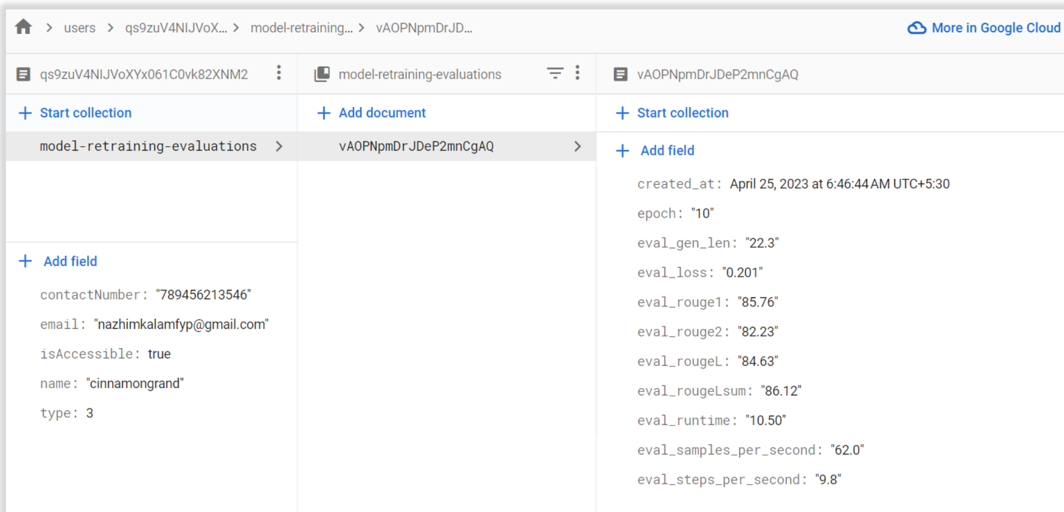
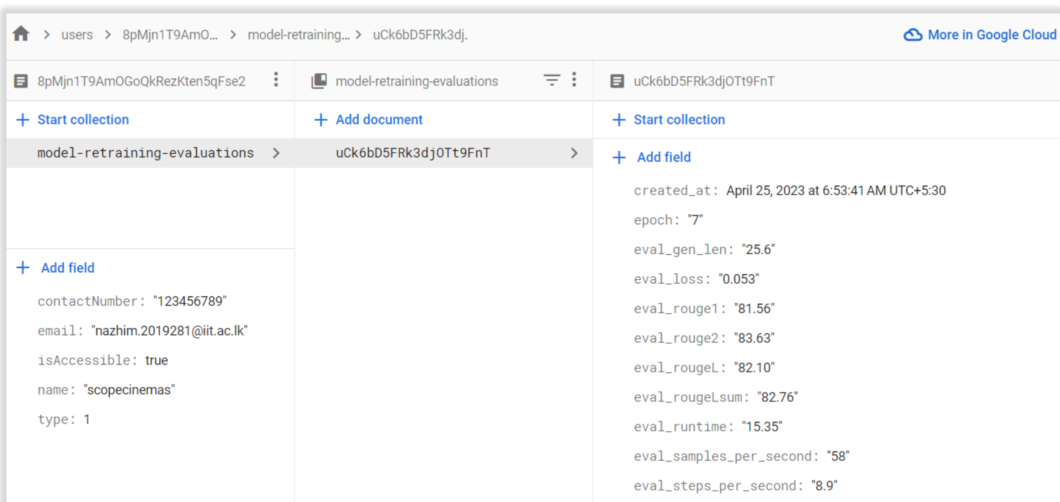


Figure 28: Movie Domain (Scope Cinema Type) Evaluation



Based on the figures presented, it is evident that the performance of the domain-specific model improves as new data is added. Therefore, each domain is expected to have its own model and continually improve it over time. The idea of generalization is considered proven based on the results presented.

8.5 Benchmarking

Previously, in **LR chapter** under the "**Benchmarking**" topic it discussed the benchmarking results of training transformers with generalized datasets for abstractive text summarization. The table given below is a benchmarking comparison with the previous researched results and the authors results.

Table 30: Benchmarking results

Year	**Type of model**	**Rouge1**	**Rouge2**	**RougeL**	**Is optimized**	**Dataset**
2019	Transformer	36.73	14.93	29.66	No	Xsum
2019	Bart	45.14	22.27	37.25	No	Xsum
2020	RoBERTa	45.42	22.13	36.92	No	Xsum
2023	Bart	80.78	79.42	80.80	Yes	Xsum

The presented table indicates the benchmarking outcomes of transformer models utilized by previous researchers between 2019 and 2020 on the same dataset (Xsum) utilized by the author. The final result displayed in the table (Year 2023) represents the author's ultimate evaluation result, which demonstrates almost a two-fold improvement in performance compared to earlier researchers who did not optimize their models.

8.6 Functional testing

The system was assessed to determine if it complies with the functional requirements outlined in **SRS Chapter** through the use of functional testing. A breakdown of the functional testing that was conducted can be found in **APPENDIX F.1**.

8.7 Module & integration testing

The high-level architecture diagram depicted in **Design Chapter** illustrated that the system's logic was divided into modules. To ensure that each module operates as intended, they underwent testing.

Table 31: Module & integration testing

Module	Input	Expected result	Actual result	Status
NLP Text parser	Triggered when input review text	Preprocessing the input data	Preprocessing the input data.	Passed
Dataset recreation	Triggered during model retraining	Decrypts data from the database and creates new dataset	Decrypts data from the database and creates new dataset	Passed
Data Encryption	Triggered when domain user generates summary	Review text gets encrypted and stored into the database	Review text gets encrypted and stored into the database	Passed
Hyperparameter tuning	Triggered during model retraining	Using the new dataset, new hyperparameters are found	Using the new dataset, new hyperparameters are found	Passed
Model training	Triggered during model retraining	Using the new dataset and new hyperparameters, model retraining is triggered	Using the new dataset and new hyperparameters, model retraining is triggered	Passed
User profile creation	Triggered when domain user firstly signup	Assigns a copy of the generalized model for the user	Assigns a copy of the generalized model for the user	Passed

8.8 Non-functional testing

The system's non-functional requirements were evaluated to assess how well they correspond with the non-functional requirements and design objectives specified in **SRS Chapter**. **APPENDIX F.2** details the specific breakdown of the non-functional testing that was executed.

8.9 Limitations of the testing process

Although the system underwent comprehensive testing, it has not been deployed or hosted yet, which makes it difficult to conduct production-grade load testing within the given timeframe.

8.10 Chapter summary

In the beginning of this chapter, the aims of the testing process and the standards for conducting it were introduced. The evaluation of the models in use was performed by testing the main research component, and functional, non-functional, and integration testing were utilized to assess the system. Ultimately, any constraints of this methodology were identified.

CHAPTER 09. EVALUATION

9.1 Chapter Overview

Once the prototype design was successfully implemented and optimized through multiple testing combinations to achieve maximum performance, the system was assessed in accordance with the requirements outlined in the **SRS chapter**. This particular chapter is devoted to the project's evaluation, which includes self-evaluation by the author and evaluations by technical, domain, and industry experts.

9.2 Evaluation Methodology & Approach

The primary focus of this research to design and implement an adaptive generalized abstractive text summarization system using optimized transformers. Although the system produces text output, it employs numerical data for decision-making, rendering it a quantitative analysis for assessing its efficacy. Additionally, a qualitative analysis by domain experts is necessary to establish its reliability. This chapter will present the results of a **thematic analysis** of the feedback obtained through the aforementioned qualitative analysis.

The demonstration video of the research, which was utilized for assessments, can be located at the following URL: https://youtu.be/2KF7PAGUxew.

9.3 Evaluation Criteria

In preparation for the evaluation process, it is essential to establish unambiguous criteria to evaluate all aspects of the research thoroughly. The following table delineates the specific criteria that the author established before initiating the evaluation.

Table 32: Evaluation criteria

CR ID	Criterion	Purpose
CR1	Selection of research domain	To confirm the importance of the selected area of study, subject matter, and gap in research.

CR2	Research contribution	To assess the importance of the discovered results and their contributions to the existing pool of knowledge.
CR3	Research documentation quality	To verify that sufficient literature has been examined and that the entire research procedure has been recorded and presented with an acceptable level of quality.
CR4	Development approach	To verify that sufficient literature has been examined and that the entire research procedure has been recorded and presented with an acceptable level of quality.
CR5	Quantitative analysis of results	To verify the metrics employed to assess and analyze the outcomes generated by the research.
CR6	Limitations & Improvements	To recognize any constraints or shortcomings and opportunities for future research.
CR7	Usability & UI/UX	To confirm that the product created for demonstration purposes is user-friendly for end-users.

9.4 Self-Evaluation

The table presented below depicts the author's self-assessment based on the aforementioned standards.

Table 33: Self-Evaluation of The Author

CR ID	**Author's self-evaluation**
CR1	The selected research domain pertained to a technical application of significant utility, which holds promise for adoption by various users across diverse domains. However, given its relatively recent introduction, identifying domain experts presents a challenge.
CR2	This research has made significant contributions in several areas aimed at enhancing the performance of text summarization using transformers. Firstly, it has made technical contributions in developing a generalized system for text summarization that can enhance domain-specific performance. Secondly, the contribution to the domain being to improve performance for review summarization for the domain of movies.

CR3	The documentation has been produced to an exceptional level of quality, with Microsoft Word being utilized for all report documentation in order to ensure consistency across diagrams, tables, and internal links along with external links (video demonstration links). Nevertheless, LaTeX has been employed for the creation of both review papers and research papers.
CR4	Considerable resources have been dedicated to gathering and preparing data to achieve optimal outcomes. Extensive experimentation has been conducted on several high-level transformer structures with diverse comprehensive datasets. Furthermore, state-of-the-art technologies and software have been employed throughout the process.
CR5	The assessment outcomes of the model were produced within the Google Colab notebooks that were utilized for training the model.
CR6	After finalizing the system, the author recognized a few areas where enhancements and improvements could be made, and suggested these as potential areas for future research.
CR7	The user interface and experience of the final product has been designed in a way that is both usable and minimalistic.

9.5 Selection of Evaluators

Evaluators were selected based on grouping, which was necessary to receive feedback on all aspects of the project. The table below illustrates the breakdown of the groups.

Table 34: Categorization of selected evaluators

CAT ID	Category
CAT1	Experts who have conducted research in the areas of Text Summarization Systems, Data Science, Data Engineering, and Machine Learning.
CAT2	Experts who possess expertise in NLP (Natural Language Processing) and transformers.
CAT3	Possible end users of the application

9.6 Evaluation Results & Expert Opinions

The opinions and feedback from the experts mentioned earlier were analyzed using thematic analysis. The results of this analysis are presented in the table below.

Table 35: Thematic analysis of expert feedback

CR ID	CAT ID	Theme	Summary of Opinions
CR1	CAT1	Text Summarization System choice gap	The utilization of text summarization can be highly advantageous for businesses, particularly in the realm of ecommerce where reviews play a crucial role in driving sales.
		Technical research gap	The adoption of transformers as an approach for the domain is preferred over conventional methods, and further investigation into its optimization can prove advantageous.
	CAT2	Domain research gap	The identified objective is to bridge the gap in performance optimization and develop a solution that is more generalized.
	CAT3	Usefulness of domain research.	As it is an adaptive solution, there currently exists no generalized application to address the performance issue in this new domain application.
CR2	CAT1	Contribution of technology to text summarization systems.	The issues with conventional approaches have been clearly identified, and there is a clear understanding of the methods required to resolve this problem.
	CAT2 & CAT3	Domain Contribution	The proposed solution is a valuable contribution since there is currently no such solution available, and creating it as a plugin for businesses could provide significant benefits.
CR3	CAT1	Displaying of content	The utilization of Microsoft Word documents to their fullest potential was acknowledged, and the

			research conducted was thorough, including the presentation of statistical data.
	CAT1 & CAT2 & CAT3	The problem-solving approach used	A comprehensive analysis of various angles was undertaken to approach the solution, and a more logical approach has been adopted.
CR4	CAT1	Data preprocessing	Significant effort has been devoted to data pre-processing, as experimentation was conducted using multiple datasets, requiring additional time and resources.
	CAT1 & CAT2	Experimentation of transformers with datasets for generalization	The selection of transformer models for experimentation was thoroughly considered and adequately justified.
	CAT1 & CAT2 & CAT3	Development approach for adaptive generalization	The chosen direction was deemed suitable by all, and a systematic and methodical approach was employed within the given timeframe.
CR5	CAT1 & CAT2 & CAT3	Analysis of the text summarization model	The current evaluation, using the ROUGE metric, indicates a significant improvement in performance compared to previous benchmarking results, with a performance gap exceeding 50%. ROUGE is considered the most suitable metric for evaluating such models.
CR6	CAT1 & CAT2	Model modifications: hybrid approach or customizing layers	The author attempted to explore the use of hybrid models and customization of existing models, but due to the limited timeframe, it was challenging to implement these approaches effectively.
	CAT3	New feature additions	Prior to generating a summary, it would be advisable to incorporate a paraphrasing function, considering the possibility that users might input text with grammatical errors.

CR7	CAT1 & CAT2	Great to have a working application for the model	As the model appears to be functioning well during testing, developing a functional application for users would be a wise choice.
	CAT 3	Minimalistic UI design	The design of the UI is very simple and clean, where most of the users found it attractive to use.

9.7 Limitations of Evaluation

As there is a scarcity of experts in the research domain (Transformers), only a limited number of expert opinions were obtained. This was apparent during the requirement gathering phase, as the author reached out to numerous ML/DL domain experts for more insights on the research concept, but only a few were able to provide relevant information. Therefore, only the experts specified in **APPENDIX G.1** possessed the required knowledge and expertise to provide constructive feedback. Nevertheless, traditional ML/DL experts were also approached to obtain feedback, as it could still prove to be beneficial.

9.8 Evaluation of Functional requirements

The breakdown of completed functional requirements is provided in **APPENDIX G.2**.

9.9 Evaluation of non-functional requirements

The breakdown of completed non-functional requirements and the attainment of the design goals are presented in **APPENDIX G.3**.

9.10 Chapter summary

In this chapter, the implemented system was evaluated by establishing evaluation criteria that comprehensively covered all aspects of the system. The author conducted self-evaluation and obtained feedback from evaluators, which was analyzed through thematic analysis. The evaluation report presented the achievement of the proposed design goals, along with the evaluation of functional and non-functional requirements.

CHAPTER 10. CONCLUSION

10.1 Chapter Overview

This chapter concludes the research project, highlighting the core functionality of the MVP's implementation. It reviews the project's achievements, obstacles encountered, and documents the author's prior knowledge and modules of the program that supported the project, along with any new knowledge and skills acquired.

10.2 Achievement of Research Aim & Objectives

10.2.1 Achievement of Aim

"*The aim of this research is to design, develop and evaluate an optimal generalized transformer architecture from a range of popularly used architectures by fine-tuning via hyperparameter optimization, therefore obtaining the recommended architecture's optimum performance.***"**

The aim of the research was achieved by **designing, developing, and evaluating** a performance adaptive generalized transformer. The core functionality was automated to meet the project requirements, and the evaluation of the work is presented in the implementation chapter.

10.2.2 Achievement of Objectives

APPENDIX H.1 lists the achievement status of the research objectives mentioned in Chapter 01, with "Completed" next to successfully completed tasks and "Incomplete" next to unfinished ones.

10.3 Utilization of knowledge from the degree

Table 36: Utilization of Knowledge Gained from The Course

Module(s)	**Utilized Knowledge**
Machine Learning	Having a solid grasp of the concept of data collection and preprocessing, as well as machine learning model training, proved to be instrumental in creating the models used in this research project.

Applied AI	The author gained a comprehensive understanding of the theoretical principles behind ML models through their knowledge of how algorithms interact during the model-building process.
Software Development Group Project	The module acted as a trial run for the Final Year Project, giving students the basic skills needed to plan, conduct and evaluate research, thus providing them with the necessary confidence and knowledge for their final year project.
Object Oriented Programming	The understanding of object-oriented programming and the significance of creating classes helped improve the development aspect of the project.
Python Programming (PP1)	This project involves the use of Flask, a web framework for the Python programming language. The PP1 module introduced the author to working with Python.
Database Systems	The understanding of using queries to communicate with the database from the webserver system facilitated read and write operations for the project.
Web Design & Development	The concepts learned from this module were used to develop the user interface (UI) of the prototype, and the foundational knowledge of using HTML, CSS, and JS was instrumental in transitioning to working with more advanced frameworks like React.

10.4 Use of Existing Skills

- **Full-Stack Web Development** – The author utilized advanced technologies for a full stack web development project while working on several R&D projects during their internship at 99x.
- **Machine Learning / Deep Learning** – During the author's internship, they were involved in several R&D projects related to data science and also utilized various online learning resources to develop their skills in machine learning.
- **Documentation Writing** – The author became proficient in project documentation during their internship and while working on the SDGP module report.

10.5 Use of New Skills

- **Text Summarization Systems –** The author lacked prior experience in text summarization systems and thus conducted research on various techniques using publicly accessible online resources from YouTube, GitHub, Google Colab, and others.
- **Data Preprocessing Techniques** – The project domain being text summarization, the author had to acquire new text preprocessing techniques to ensure the data was meaningful.
- **Hyperparameter Optimization** – The author explored and experimented with hyperparameter tuning frameworks to automate the hyperparameter search, referring to tutorials and technical articles to implement this in the project.

10.6 Achievement of learning outcomes

In **APPENDIX H.2**, there is a presentation of the accomplishment of the learning objectives.

10.7 Problems and Challenges faced

Table 37: Mitigations to Problems and Challenges Faced

Problem/Challenge	**Mitigation**
Significant training time and computing resource limitations.	The author uses Google Colab to train transformer-based models due to the high GPU power demands of the training process.
Limited experts for the domain	The author conducted interviews with domain experts for requirement gathering by reaching out to them via LinkedIn since there were limited opportunities for in-person interactions with domain experts.
Power outages were frequent, causing battery and internet connectivity issues.	The author persevered with the project despite power outages by working late or early in the morning at co-working spaces.

10.8 Deviations

The author's original goal was to develop an optimized solution for ***movie review summarization*** using transformers. However, after discussions with supervisors, it was decided that the author's technical contribution of automated hyperparameter tuning for transformers was not significant enough. As a result, the idea of creating a ***performance adaptive generalized solution*** was considered to continue the research implementation. The Gantt chart plan at the beginning and end of the project can be located in **APPENDIX H.3.**

It's great to see that the author was able to complete the core functionalities for the prototype, even with the challenge of time constraints mentioned by the experts interviewed for the requirement gathering. It shows the author's determination and ability to overcome obstacles to achieve the project's goals.

10.9 Limitations of the research

- The author tried to implement additional performance improvements like model hybridization & model layer customization after completing the core implementation. However, due to the limited time available, the amount of research required for transformer hybridization was significant, which prevented the author from continuing.
- The author did not explore various other transformer models for abstractive text summarization due to limitations of GPUs.

10.10 Future enhancements

- It’s suggested that transformer hybridization/ model layer customization can be used to improve the performance of text summarization models in the future.
- The author suggests that including text paraphrase models for user reviews would be a sensible approach since there is a potential that user reviews entered are not always accurate.

- The author suggests applying key word extraction for the sentiment classification of the review summary in order to identify which key words contributed to the sentiment classification and to help domain users improve their service.

10.11 Achievement of the Contribution to the Body of Knowledge

The author made successful contributions to the problem domain of movie review summarization by implementing the system and making deviations from the initial goal to create a generalized solution. They also made technical contributions to improve the system's performance, but due to limited time, they were unable to explore all possibilities. Finally, the author made additional contributions to bring the research project to a conclusion.

10.11.1 Domain Contributions

The author was able to address the performance gap listed for movie review summarization, in order for the need of advanced approaches to increase the performance and achieve a better result.

Moreover, generalization approach considered here contributions to various other domains facing the similar problem to be address as a common.

10.11.2 Technical Contribution

Using a top-tier explored transformer model, automating hyperparameter search for every domain, and use the newly exposed data to automate model retraining with the searched optimal set of hyperparameters which is adaptive with respect to the domain.

Currently, there are no such approaches taken from the research done by the author, and the author believes that this approach would benefit multiple domains at the same time.

10.11.3 Additional Contributions

1. Research-based Data Preprocessing scripts specifically for text summarization issue domains.
2. Sentiment Analysis on the generated review summary
3. Experimented the model training with multiple datasets, to get the best possible set of evaluation results.
4. Integrating email notifications to inform domain users about the status of model retraining.

10.12 Implementation code

All related code and documentation material are made available in GitHub by the author at **https://github.com/nazhimkalam/gensum**

10.13 Concluding remarks

The study designed, built, and evaluated an adaptive generalized abstractive text summarization system using optimized transformers and automated hyperparameter tuning and model retraining. The author met the goals and objectives of the project and discussed the role of their prior knowledge and academic background in supporting the research. The author acquired new skills during the project and faced challenges and obstacles. They discussed deviations taken and limitations of the research, as well as opportunities for future improvements. The research contributed to the body of knowledge with domain, technical, and additional contributions made. The author plans to publish a research paper based on their findings.

REFERENCES

Abolghasemi, M., Dadkhah, C. and Tohidi, N. (2022). HTS-DL: Hybrid Text Summarization System using Deep Learning. 2022 27th International Computer Conference, Computer Society of Iran (CSICC). 23 February 2022. Tehran, Iran, Islamic Republic of: IEEE, 1–5. Available from https://doi.org/10.1109/CSICC55295.2022.9780395 [Accessed 26 October 2022].

Alsaqer, A.F. and Sasi, S. (2017). Movie review summarization and sentiment analysis using rapidminer. 2017 International Conference on Networks & Advances in Computational Technologies (NetACT). July 2017. Thiruvanthapuram, India: IEEE, 329–335. Available from https://doi.org/10.1109/NETACT.2017.8076790 [Accessed 10 October 2022].

Barna, N.H. and Heickal, H. (2022). An Automatic Abstractive Text Summarization System. Dhaka University Journal of Applied Science and Engineering, 6 (2), 39–48. Available from https://doi.org/10.3329/dujase.v6i2.59217.

Boorugu, R., Ramesh, G. and Madhavi, K. (2019). Summarizing Product Reviews Using Nlp Based Text Summarization. International Journal of Scientific & Technology Research Volume, 8 (10), 1127–1133.

Brasoveanu, A.M.P. and Andonie, R. (2020). Visualizing Transformers for NLP: A Brief Survey. 2020 24th International Conference Information Visualisation (IV). September 2020. Melbourne, Australia: IEEE, 270–279. Available from https://doi.org/10.1109/IV51561.2020.00051 [Accessed 2 November 2022].

Etemad, A.G., Abidi, A.I. and Chhabra, M. (2021). A Review on Abstractive Text Summarization Using Deep Learning. 2021 9th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO). 3 September 2021. Noida, India: IEEE, 1–6. Available from https://doi.org/10.1109/ICRITO51393.2021.9596500 [Accessed 10 October 2022].

Gupta, A. et al. (2021). Automated News Summarization Using Transformers. ArXiv, abs/2108.01064.

Gupta, V. and Lehal, G.S. (2010). A Survey of Text Summarization Extractive Techniques. Journal of Emerging Technologies in Web Intelligence, 2 (3), 258–268. Available from https://doi.org/10.4304/jetwi.2.3.258-268.

Joy, J. and Selvan, M.P. (2022). A comprehensive study on the performance of different Multi-class Classification Algorithms and Hyperparameter Tuning Techniques using Optuna. 2022 International Conference on Computing, Communication, Security and Intelligent Systems (IC3SIS). 23 June 2022. Kochi, India: IEEE, 1–5. Available from https://doi.org/10.1109/IC3SIS54991.2022.9885695 [Accessed 24 October 2022].

Khan, A. et al. (2020). Movie Review Summarization Using Supervised Learning and Graph-Based Ranking Algorithm. Computational Intelligence and Neuroscience, 2020, 7526580. Available from https://doi.org/10.1155/2020/7526580.

Kirmani, M. et al. (2019). Hybrid Text Summarization: A Survey. In: Ray, K. Sharma, T.K. Rawat, S. et al. (eds.). Soft Computing: Theories and Applications. Advances in Intelligent Systems and Computing. Singapore: Springer Singapore, 63–73. Available from https://doi.org/10.1007/978-981-13-0589-4_7 [Accessed 1 November 2022].

Kouris, P., Alexandridis, G. and Stafylopatis, A. (2019). Abstractive Text Summarization Based on Deep Learning and Semantic Content Generalization. Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics. 2019. Florence, Italy: Association for Computational Linguistics, 5082–5092. Available from https://doi.org/10.18653/v1/P19-1501 [Accessed 24 October 2022].

Lackermair, G., Kailer, D. and Kanmaz, K. (2013). Importance of Online Product Reviews from a Consumer's Perspective. Advances in Economics and Business, 1 (1), 1–5. Available from https://doi.org/10.13189/aeb.2013.010101.

Lin, C.-Y. (2004). ROUGE: A Package for Automatic Evaluation of Summaries. 8.

Liu, X. and Wang, C. (2021). An Empirical Study on Hyperparameter Optimization for Fine-Tuning Pre-trained Language Models. Available from http://arxiv.org/abs/2106.09204 [Accessed 24 October 2022].

Mahajan, R. et al. (2021). Text Summarization Using Deep Learning. International Research Journal of Engineering and Technology (IRJET), 08 (05th May 2021), 1737–1740.

McAuley, J.J. and Leskovec, J. (2013). From amateurs to connoisseurs: modeling the evolution of user expertise through online reviews. Proceedings of the 22nd international conference on World Wide Web - WWW ’13. 2013. Rio de Janeiro, Brazil: ACM Press, 897–908. Available from https://doi.org/10.1145/2488388.2488466 [Accessed 19 November 2022].

Mukherjee, R. et al. (2020). Read what you need: Controllable Aspect-based Opinion Summarization of Tourist Reviews. Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval. 25 July 2020. 1825–1828. Available from https://doi.org/10.1145/3397271.3401269 [Accessed 10 October 2022].

Neyshabur, B. et al. (2017). Exploring Generalization in Deep Learning. undefined. Available from https://www.semanticscholar.org/reader/d53fb3feeeab07a0d70bf466dd473ec6052ecc07 [Accessed 9 November 2022].

Pai, A. (2014). Summarizer Using Abstractive and Extractive Method. International Journal of Engineering Research, 3 (5), 5.

Pizam, A. and Ellis, T. (1999). Customer satisfaction and its measurement in hospitality enterprises. International Journal of Contemporary Hospitality Management, 11 (7), 326–339. Available from https://doi.org/10.1108/09596119910293231.

Shi, T. et al. (2020). Neural Abstractive Text Summarization with Sequence-to-Sequence Models. Available from http://arxiv.org/abs/1812.02303 [Accessed 10 October 2022].

Shorten, C. and Khoshgoftaar, T.M. (2019). A survey on Image Data Augmentation for Deep Learning. Journal of Big Data, 6 (1), 60. Available from https://doi.org/10.1186/s40537-019-0197-0.

Socher, R., Bengio, Y. and Manning, C.D. (2012). Deep Learning for NLP (without Magic). Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Tutorial Abstracts. July 2012. Jeju Island, Korea: Association for Computational Linguistics, 5. Available from https://aclanthology.org/P12-4005 [Accessed 2 November 2022].

Steinberger, J. and Jezek, K. (2009). Evaluation Measures for Text Summarization. Comput. Informatics, 28 (2), 251–275.

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. 2020. Online: Association for Computational Linguistics, 38–45. Available from https://doi.org/10.18653/v1/2020.emnlp-demos.6 [Accessed 10 October 2022].

Zhang, J. et al. (2020). PEGASUS: Pre-training with Extracted Gap-sentences for Abstractive Summarization. Available from http://arxiv.org/abs/1912.08777 [Accessed 18 October 2022].

Zhou, K. et al. (2021). Domain Generalization with MixStyle. undefined. Available from https://www.semanticscholar.org/reader/4f6eafafc9563a5b904535078df7e74afe39ef59 [Accessed 5 December 2022].

aws.amazon.com. (2022). Hyperparameter optimization for fine-tuning pre-trained transformer models from Hugging Face | AWS Machine Learning Blog. [online] Available at: https://aws.amazon.com/blogs/machine-learning/hyperparameter-optimization-for-fine-tuning-pre-trained-transformer-models-from-hugging-face/ [Accessed 9 Nov. 2022].

Web data: Amazon Movie Reviews (no date) SNAP. Available at: https://snap.stanford.edu/data/web-Movies.html (Accessed: April 25, 2023).

Jonathanoheix (2018) Sentiment analysis with Hotel Reviews, Kaggle. Kaggle. Available at: https://www.kaggle.com/code/jonathanoheix/sentiment-analysis-with-hotel-reviews/input (Accessed: April 25, 2023).

Najmuddin, S., Atal, Z. and Ziar, R.A. (1970) [PDF] comparative analysis of power consumption of the linux and its distribution operating systems vs windows and mac operating systems.: Semantic scholar, Kardan journal of engineering and technology. Available at: https://www.semanticscholar.org/paper/Comparative-Analysis-of-Power-Consumption-of-the-vs-Najmuddin-Atal/ae640a6873317596bd8869f0a2155e7bc4d5e899 (Accessed: April 25, 2023).

McKinney, W. (2017) Python for Data Analysis: Data wrangling with pandas, NumPy, and ipython, Google Books. "O'Reilly Media, Inc.". Available at:

https://books.google.com/books/about/Python_for_Data_Analysis.html?id=BCc3DwAAQBAJ (Accessed: April 25, 2023).

Chauhan, N. et al. (1970) [PDF] implementation of database using Python Flask Framework: Semantic scholar, International Journal of Engineering and Computer Science. Available at: https://www.semanticscholar.org/paper/Implementation-of-database-using-python-flask-Chauhan-Singh/7de6ad96c6f431cc9084be0a0d788149ad267ffc (Accessed: April 25, 2023).

Paszke, A. et al. (1970) [PDF] automatic differentiation in pytorch: Semantic scholar, [PDF] Automatic differentiation in PyTorch | Semantic Scholar. Available at: https://www.semanticscholar.org/paper/Automatic-differentiation-in-PyTorch-Paszke-Gross/b36a5bb1707bb9c70025294b3a310138aae8327a (Accessed: April 25, 2023).

Canesche, M. et al. (1970) Google colab CAD4U: Hands-on cloud laboratories for digital design: Semantic scholar, 2021 IEEE International Symposium on Circuits and Systems (ISCAS). Available at: https://www.semanticscholar.org/paper/Google-Colab-CAD4U%3A-Hands-On-Cloud-Laboratories-for-Canesche-Bragan%C3%A7a/6d0cbb5d66d8dc80a447d68b86fa1bddf0498494 (Accessed: April 25, 2023).

Virtanen, P. et al. (1970) [PDF] scipy 1.0: Fundamental algorithms for scientific computing in python: Semantic scholar, Nature Methods. Available at: https://www.semanticscholar.org/paper/SciPy-1.0%3A-fundamental-algorithms-for-scientific-in-Virtanen-Gommers/f0d35b37fec26c3f1ed09253cbb9304fb62208d1 (Accessed: April 25, 2023).

Abadi, M. et al. (1970) [PDF] tensorflow: A system for large-scale machine learning: Semantic scholar, ArXiv. Available at: https://www.semanticscholar.org/paper/TensorFlow%3A-A-system-for-large-scale-machine-Abadi-Barham/46200b99c40e8586c8a0f588488ab6414119fb28 (Accessed: April 25, 2023).

Paszke, A. et al. (1970) [PDF] pytorch: An imperative style, high-performance deep learning library: Semantic scholar, [PDF] PyTorch: An Imperative Style, High-Performance Deep Learning Library | Semantic Scholar. Available at: https://www.semanticscholar.org/paper/PyTorch%3A-An-Imperative-Style%2C-High-Performance-Deep-Paszke-Gross/3c8a456509e6c0805354bd40a35e3f2dbf8069b1 (Accessed: April 25, 2023).

Waranashiwar, J. and Ukey, M. (1970) Ionic framework with angular for hybrid app development, Semantic Scholar. Available at: https://www.semanticscholar.org/paper/Ionic-Framework-with-Angular-for-Hybrid-App-Waranashiwar-Ukey/92847c13951213e701e5b3a834e7e45b5c873644 (Accessed: April 25, 2023).

Wahyudi, E.T., Erwin, A. and Lim, C. (1970) [PDF] development of API middleware and mobile application for a job marketplace by using restful API and mobile development framework: Semantic scholar, [PDF] Development of API Middleware and Mobile Application for a Job marketplace by Using RESTful API and Mobile Development Framework | Semantic Scholar. Available at: https://www.semanticscholar.org/paper/Development-of-API-Middleware-and-Mobile-for-a-Job-Wahyudi-Erwin/03166228259cd38cc5c433b77e7d2510fed72764 (Accessed: April 25, 2023).

Verma, A. et al. (1970) Web application implementation with Machine Learning: Semantic scholar, 2021 2nd International Conference on Intelligent Engineering and Management (ICIEM). Available at: https://www.semanticscholar.org/paper/Web-Application-Implementation-with-Machine-Verma-Kapoor/9f9dc62da64b368e2503eb312b8ef9b687bc5b1a (Accessed: April 25, 2023).

Relan, K. (1970) Deploying flask applications: Semantic scholar, Building REST APIs with Flask. Available at: https://www.semanticscholar.org/paper/Deploying-Flask-Applications-Relan/f55aa0f7c950e7471068e5fe9f72f6221ca6e3f0 (Accessed: April 25, 2023).

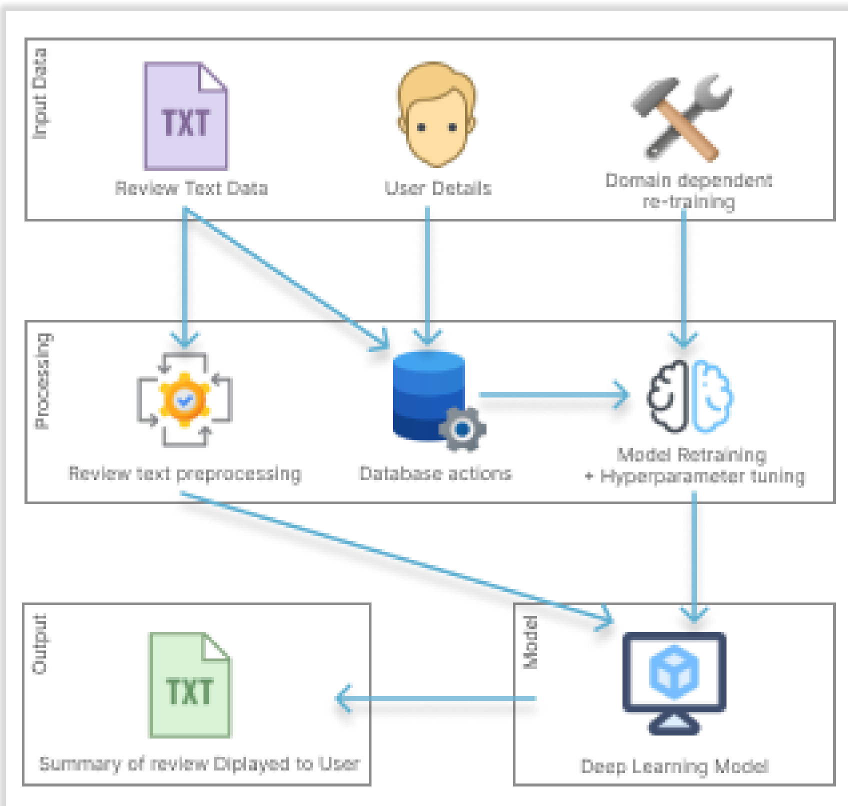
Srivastava, A. (1970) Django, the python web framework: Semantic scholar, INTERANTIONAL JOURNAL OF SCIENTIFIC RESEARCH IN ENGINEERING AND MANAGEMENT. Available at: https://www.semanticscholar.org/paper/Django-%2C-The-Python-Web-Framework-Srivastava/231c909120972252411f85b4bb8c13657ede82c6 (Accessed: April 25, 2023).

APPENDIX A – INTRODUCTION

A.1. Prototype feature diagram

The illustration presented underneath illustrates the preliminary characteristic diagram suggested in the proposal manuscript.

Figure 29: Prototype Feature Diagram (Self-Composed)



A.2. Project scope

In scope

- ***System Generalization*** – Creating a generalized system to be able to **adapt** to any domain.
- ***Dataset reconstruction*** – Reconstructing the dataset to a format structure which can be used for data preprocessing + model training.
- ***Model refinement on hyperparameter tuning*** – Performing hyperparameter tuning on the top tier transformer architecture models.

- ***Evaluating the models*** – Evaluating all the architectures using appropriate metrics to filter out the best architecture from the rest.
- ***API integration development*** – REST API endpoints will be created to serve/call the final chosen model for interactions.
- ***GUI development*** – A graphical user interface will be developed; therefore, the end user will be able to perform abstractive text summarization and get visual results.

Out scope

- ***Limited architecture explored*** **–** The system will only be explored with few of the top tier architectures (roughly around 3 or 5 maximum), and will not be exploring more than that.
- ***Only single model integration*****–** The final model which outperforms the rest with the best set of hyperparameters will be used as the summary generation model, options to select other architectures explored with their hyperparameters aren't included.

APPENDIX B – LITERATURE REVIEW

B.1. Related work in abstractive text summarization

Table 38: Related Work in Abstractive Text Summarization

Ref.	**Summary**	**Contribution**	**Limitations**
(Khan, Gul, Zareei, et al., 2020)	An automatic approach to summarize lengthy movie reviews and allow users to quickly recognize the positive and negative aspects of a movie.	Worked on feature extraction and converting reviews into vector space, followed by the Naïve Bayes machine learning algorithm used for review classification, using an undirected weighted graph based ranking algorithm to rank score for reach review sentence in graph. Finally, the top ranked sentences are chosen based on highest rank scores to produce ***extractive summary***.	To use advanced deep learning approaches.
(Boorugu, Ramesh and Madhavi, 2019)	Using customer reviews on products when making purchasing decisions to give a proper summarization of the reviews to the customer, so that he doesn't need to go through all the reviews to figure out if the product is what he is looking for and save time.	Using seq2seq model for summarization along with attention mechanism for increased accuracy, also using word embedding model Concept net Number batch which is better than Glove. Finally, using a 1D convolutional layer followed by max pooling layer, LSTM layer and then at the end a fully connected layer.	Focused on improving the accuracy by using the latest models in the field of text summarization. By using transformers

			architecture, we could improve this.
(Mukherjee et al., 2020)	A solution for generating personalized aspect-based opinion summaries from large collections of online tourist reviews, also able to customize the attributes of the summary based on the user's interest.	Using an Integer Linear Programming (ILP [Unsupervised method]) based extractive technique to select an informative subset of opinions around the identified aspects. Evaluate and compare the summaries using ROUGE based metrics and obtain competitive results.	Motive for the need to create tourist review dataset for our experiments. The need for also experimenting with the data of lesser known places (Tourist locations)
(Gupta et al., 2021)	A comprehensive comparison of a few transformer architecture based pre-trained models for text summarization.	Using the pretrained models such as Pipeline BART, BART modified, T5 and PEGASUS to work with the text summarization. Evaluation metrics we done using the ROUGE Scores.	Future work should focus on building more robust models which can further extend the algorithm to create summaries of variable length and apply for multi-document summarization.

(Mahajan et al., 2021)	Generate a text summary along with proper grammar and no repeated words using the Encoder-Decoder model with the attention layer	Developed an encoder-decoder model using Gated Recurrent Units and trained the model to generate abstractive summary from an article.	Real time training required if this is used in production, in order to train with the latest articles with time.
(Etemad, Abidi and Chhabra, 2021)	Experimenting the text summarization domain with deep learning approaches and finding which performs the best, from RNN, CNN, Transformers etc.…	Experimenting with RNN based models' architectures, working with pre-trained transformer-based model architectures. Finally, using evaluation metrics such as BLEU and ROUGE to evaluate the models	NA

APPENDIX C – SRS

C.1. Requirement Elicitation Methodologies

Table 39: Stakeholder groups (*Self-Composed*)

Group	Stakeholders	Reason	Instrument
G1	Domain experts (NLP Experts, AI Researchers, Data Scientists)	In order to respond to research questions and discover anything the author may have overlooked, gather any insights and information especially in the study area.	Interview
G2	Domain and General Users	Gather requirements which will help develop features expected in the application.	Survey & LR
G3	Competitors	Analyze any existing systems related to the research and understand how the project can be enhanced	Self-Evaluations & Brain Storming
G4	Developers	Cross checking if the project is feasible to be continued with.	Prototyping

C.2. Interview analysis

Table 40: Interview thematic analysis (*Self-Composed*)

Code	Theme
Data handling	Data Collection & Data Preprocessing
Transformer architectures	Best performing transformer architectures
Generalization	Handling adaptive generalization
Research scope	Research gap and scope
Hyperparameter tuning	Automatic hyperparameter tuning & model retraining
Hybrid transformers	Looking into hybrid transformer combinations
Custom transformers	Customizing the transformer architecture

Prototype	Prototype features and suggestions
Business benefits	Understanding which and how businesses would benefit
Evaluations	Understanding the importance and evaluation ways

Table 41: Interview participant details

ID	Affiliation	Expertise related to the research
P1	PhD Research Student in Computational Linguistics	NLP
P2	Machine Learning Expertise | Lecturer with PhD	ML and Neural Networks
P3	NLP Researcher	NLP
P4	Software Architect	Algorithms
P5	Software Architect & ML Researchers	ML & Algorithms
P6	VP Innovations, Software Engineer	ML & Neural Networks
P7	Lecturer with MSc	NLP

Table 42: Interview thematic analysis themes, conclusions & evidence

Theme	Conclusion
Data handling	Since this is a project connected to data science, the availability of data and the data preparation methods to be used are the main concerns. PhD candidates suggested to make use of verified and well researched datasets for the area of generalization since every domain will be using the same model initially to start off with, therefore the quality of data should be considered, it was recommended to use datasets that have already been studied and utilized by other researchers since they have done so and verified their findings. NLP researches were concerned on the language of text the project scope is into when performing text preprocessing, since text data can also contain other language characters unless the project is scoped down to only English language supportive.

Transformer architectures	Most of the interviewees pointed out similar transformers architectures which they have used and found impressive results, which are mostly BERT, GPT-2, Roberta, T5 etc... where they have explored not only with text summarization but also when other NLP areas such as sentiment analysis, proving again that transformers are well known for solving NLP problems. They also stated to check up with the daily stats (most downloads and likes) about the transformer architectures from Hugging Face, this is because new better versions of the transformers are always been produced/updated.
Generalization	The Software Engineers and Architects suggested to make use of document-oriented NoSQL database management system to handling data storage for the domain specific managers, this is because its easily scalable and provider superior performance especially for the idea of adaptative generalization for this project. Such services are like MongoDB, Firebase NoSQL DB etc.
Research scope	The technology exports and research experts find that the solution of solving this problem using optimized transformers is great but they find that creating a generalized adaptive solution would be challenging with the time frame of the project but also advised to solve for the domain of movies first and then get into the others if time permits.
Hyperparameter tuning	The NLP researchers and Lectures suggested several ways of using tools and libraries to help with hyperparameter tuning since doing this manually is very time consuming and unnecessary effort.
Hybrid transformers	PhD candidates liked the idea of using hybrid transformer combination by using ensemble approaches to combine the top best two transformer architecture but it seems the scope of the project for the time frame is becoming bigger and riskier.
Custom transformers	The NLP researchers recommended to customize the existing transformer architecture instead of Hybrid model creation because of the project scope.
Prototype	The interviewees are interested to see how the generalization system for domain specific retraining is going to work together since they haven’t seen any such approach earlier from their experience. They also suggested if time

	permits to make use of a pretrained model to get the sentiment of the summary aswell to be displayed on the GUI.
Business benefits	Most of the interviewees suggested the Movie domain, Tourism, Ecommerce, Book, Researchers would find this useful in summarizing their customer reviews on their businesses.
Evaluations	The PhD candidates and NLP experts suggested the importance of evaluations when it comes to dealing with the adaptive generalization model since this can be used in any domain, therefore suggesting the author of the project to explore maximum of 3 domains when working with so its easier to compare the evaluation results else it will be confusing when demonstrating the work to anyone.

C.3. Use case descriptions

Table 43: Usecase mappings (*Self-Composed*)

Use case Id	**Use case name**
UC01	Input Review
UC02	Create Profile
UC03	Retrain Model
UC04	Search New Hyperparameters
UC05	Create Model
UC06	Prepare Dataset
UC07	View Summary
UC08	Generate Summary
UC09	Store Data
UC10	Delete reviews

Table 44: Use case description UC:01 (*Self-Composed*)

Use Case Name	Input Review

Use Case Id	UC:01
Description	Requested the user to input a text review
Primary Actor	General User, Domain Specific User
Pre-Conditions	Domain Specific user needs to be login in before this action
Extended use cases	None
Included use cases	None
Trigger	A user selects the text input field to enter text review.
Main flow	The general user clicks on the input field to enter the review text, if it's a domain specific user then user needs to login into the application for this action
Alternative flows	None
Expectational flows	Displays an error message if the network request fails (server is down, or internet issues from client).
Post Conditions	None

Table 45: Use case description UC:02 (*Self-Composed*)

Use Case Name	Create Profile
Use Case Id	UC:02
Description	Domain users will be able to create a unique profile to manage their content
Primary Actor	Domain Specific User
Pre-Conditions	None
Extended use cases	None
Included use cases	None

Trigger	The domain user signups an account with in the system	
Main flow	Actor	System
	1. The domain user navigates to the sign-in page. 2. The domain user clicks on sign in, to register their self or login to the application	3. Create a new user in the database and notify the user.
Alternative flows	None	
Expectational flows	Displays an error message if the network request fails (server is down, or internet issues from client).	
Post Conditions	Success message displayed.	

Table 46: Use case description UC:10 (*Self-Composed*)

Use Case Name	Delete reviews	
Use Case Id	UC:10	
Description	Domain users will only be able to perform this action to manage their own data reviews and delete	
Primary Actor	Domain Specific User	
Pre-Conditions	Domain user should be logged into the application	
Extended use cases	None	
Included use cases	None	
Trigger	Clicking on the delete action button on the review card list	
Main flow	Actor	System
	1. The domain user logins into the application	4. Searches for the review with the user id and the review id on the database.

	2. Navigates to the manage reviews area 3. Clicks on ‘Delete’ on the choice of review by the domain user	5. Deletes the review from the database.
Alternative flows	None	
Expectational flows	Displays an error message if the network request fails (server is down, or internet issues from client).	
Post Conditions	Success message displayed.	

Table 47: Use case description UC:04 (*Self-Composed*)

Use Case Name	Search new hyperparameters
Use Case Id	UC:04
Description	Searching for new set of hyperparameters during model retraining process.
Primary Actor	Domain Specific User
Pre-Conditions	Domain user should have entered enough data into the system
Extended use cases	None
Included use cases	Retrain Model
Trigger	Domain user have triggered the model retraining from the UI by clicking on to the “Retrain model” button
Main flow	1. New unseen data is fetched from the database. 2. The data is used for automated hyperparameter model training. 3. New hyperparameter is used for model training
Alternative flows	None
Expectational flows	None

Post Conditions	None

Table 48: Use case description UC:05 (*Self-Composed*)

Use Case Name	Create model
Use Case Id	UC:05
Description	Using the new set of hyperparameters found the model is retrained to create a new updated version
Primary Actor	Domain Specific User
Pre-Conditions	Domain user should have entered enough data into the system
Extended use cases	None
Included use cases	Retrain Model
Trigger	Domain user have triggered the model retraining from the UI by clicking on to the “Retrain model” button
Main flow	1. Newly found hyperparameters are used to retrain the model. 2. Old model is replaced with the new model.
Alternative flows	None
Expectational flows	None
Post Conditions	None

Table 49: Use case description UC:06 (*Self-Composed*)

Use Case Name	Prepare dataset
Use Case Id	UC:06
Description	Pulling the new data from the database in order to create a new dataset for model retraining
Primary Actor	Domain Specific User

Pre-Conditions	Domain user should have entered enough data into the system
Extended use cases	None
Included use cases	Retrain Model
Trigger	Domain user have triggered the model retraining from the UI by clicking on to the "Retrain model" button
Main flow	1. Gets the parameters sent from the request body. 2. Fetches data from the database related to the parameters. 3. Creating new dataset using the data.
Alternative flows	None
Expectational flows	None
Post Conditions	None

Table 50: Use case description UC:08 (*Self-Composed*)

Use Case Name	Generate Summary	
Use Case Id	UC:08	
Description	Generating summary for the input review using the latest model saved.	
Primary Actor	Domain Specific User, General User	
Pre-Conditions	User should have entered a review text from the frontend to generate a summary for.	
Extended use cases	None	
Included use cases	View Summary	
Trigger	User clicked on "Generate summary" after using the review text as input.	
Main flow	Actor	System

	1. User should have entered a text from the frontend in the input field requested. 2. User clicks on "General summary"	3. System uses the input review to perform data preprocessing. 4. System uses the preprocessed text review to generate the summary
Alternative flows	None	
Expectational flows	Displays an error message if the network request fails (server is down, or internet issues from client).	
Post Conditions	Success message displayed.	

Table 51: Use case description UC:09 (*Self-Composed*)

Use Case Name	Store data	
Use Case Id	UC:09	
Description	Storing the review and summary data along with the sentiment.	
Primary Actor	Domain Specific User	
Pre-Conditions	Domain user should have entered input review and requested	
Extended use cases	None	
Included use cases	View summary	
Trigger	Domain user clicks on 'Generate summary' after adding a review text	
Main flow	Actor	System
	1. User should have entered a text from the frontend in the input field requested. 2. User clicks on "General summary"	3. The review data is used to generate the summary. 4. Using the generated summary to get the

		sentiment and sentiment score. 5. The result of all these will be written into the database
Alternative flows	None	
Expectational flows	None	
Post Conditions	None	

C.4. Functional requirements

Table 52: ‘MoSCoW’ technique of requirement prioritization

Priority level	**Description**
Must have (M)	The demand at this level is the fundamental functional requirement for a prototype, and it must be carried out.
Should have (S)	Although not strictly required for the anticipated prototype to function, important criteria do provide a lot of value.
Could have (C)	Optional, non-essential desirable needs are crucial to the project's scope.
Will not have (W)	Requirements that the system might not meet right now and that are not given first consideration.

APPENDIX D – DESIGN

D.1. UI wireframes

Figure 30: Homepage Wireframe (Self-Composed)

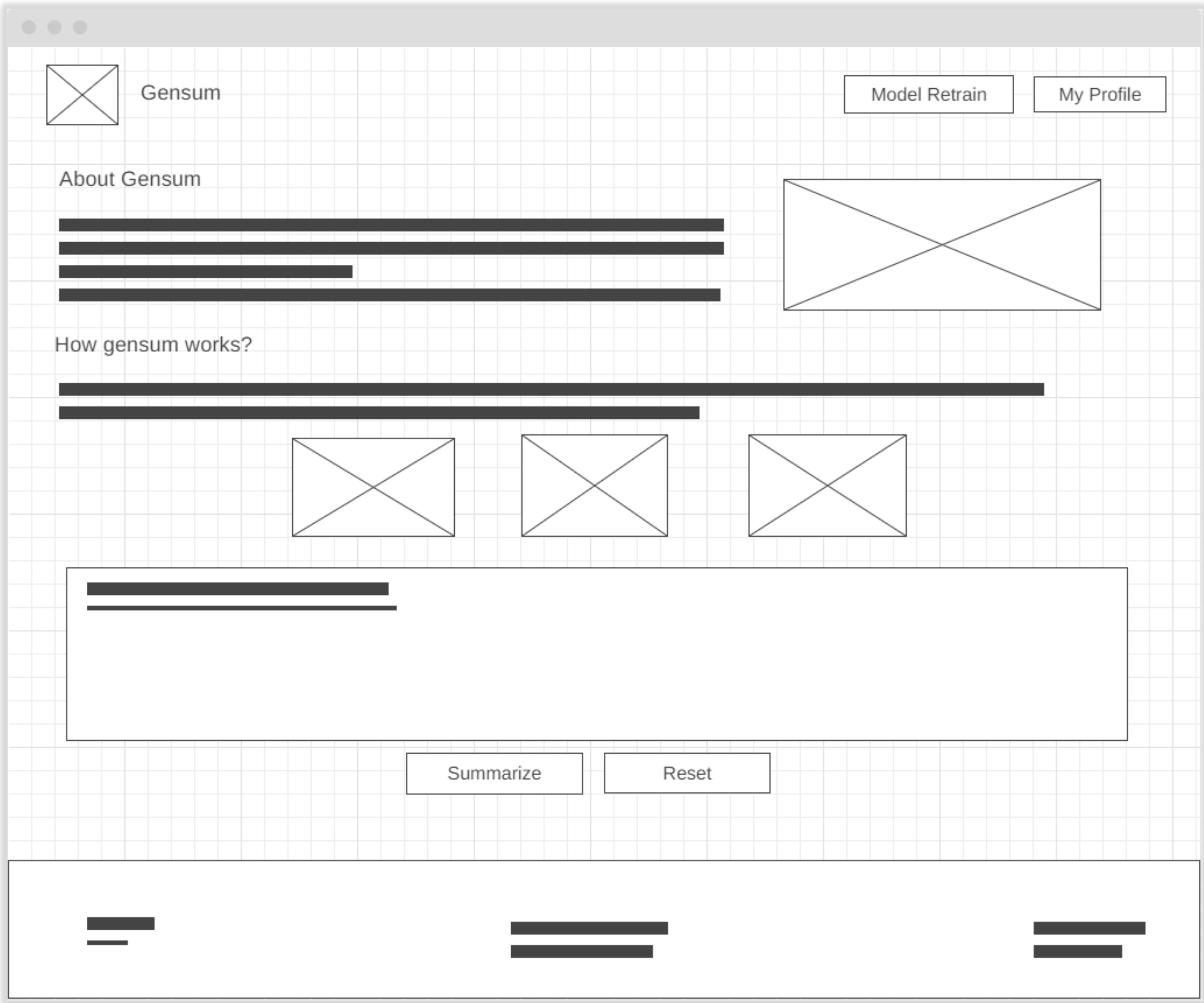


Figure 31: Review History Page (Self-Composed)

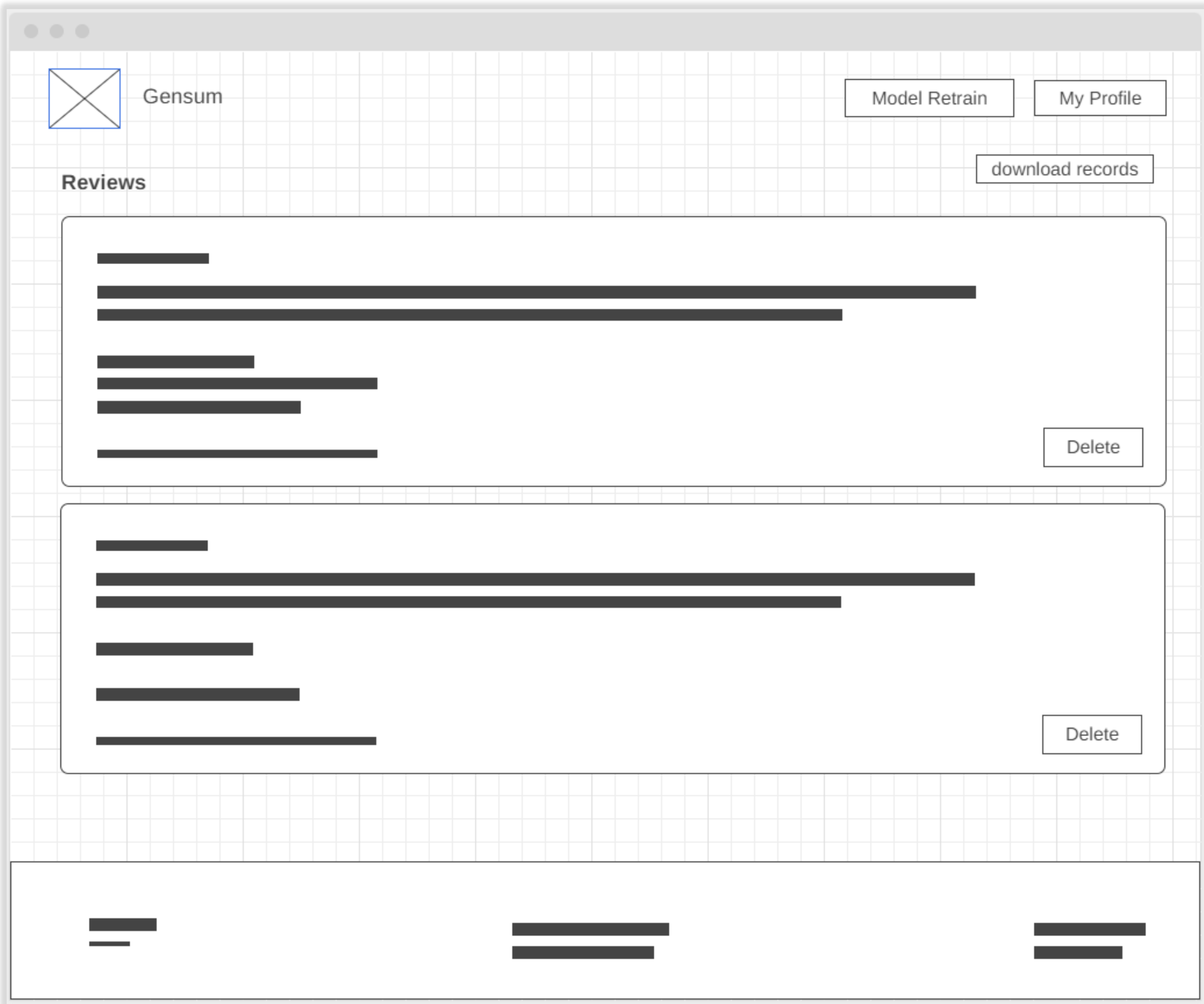


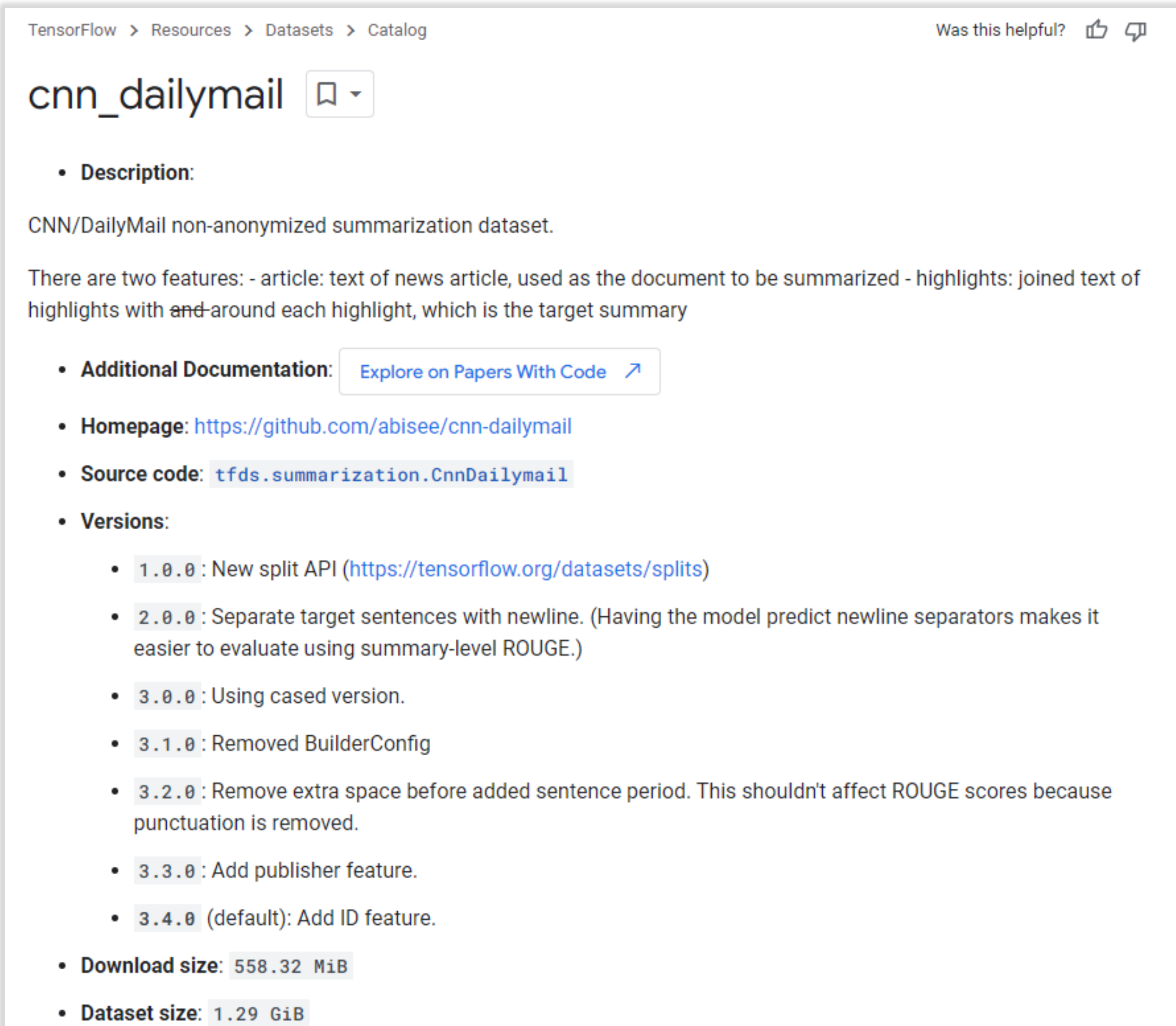
Figure 32: User Profile Page (Self-Composed)

APPENDIX E – IMPLEMENTATION

E.1. Dataset Resources

- **CNN Dailymail**

Figure 33: CNN Dailymail Dataset (*view*)



TensorFlow > Resources > Datasets > Catalog

Was this helpful?

cnn_dailymail

- **Description**:

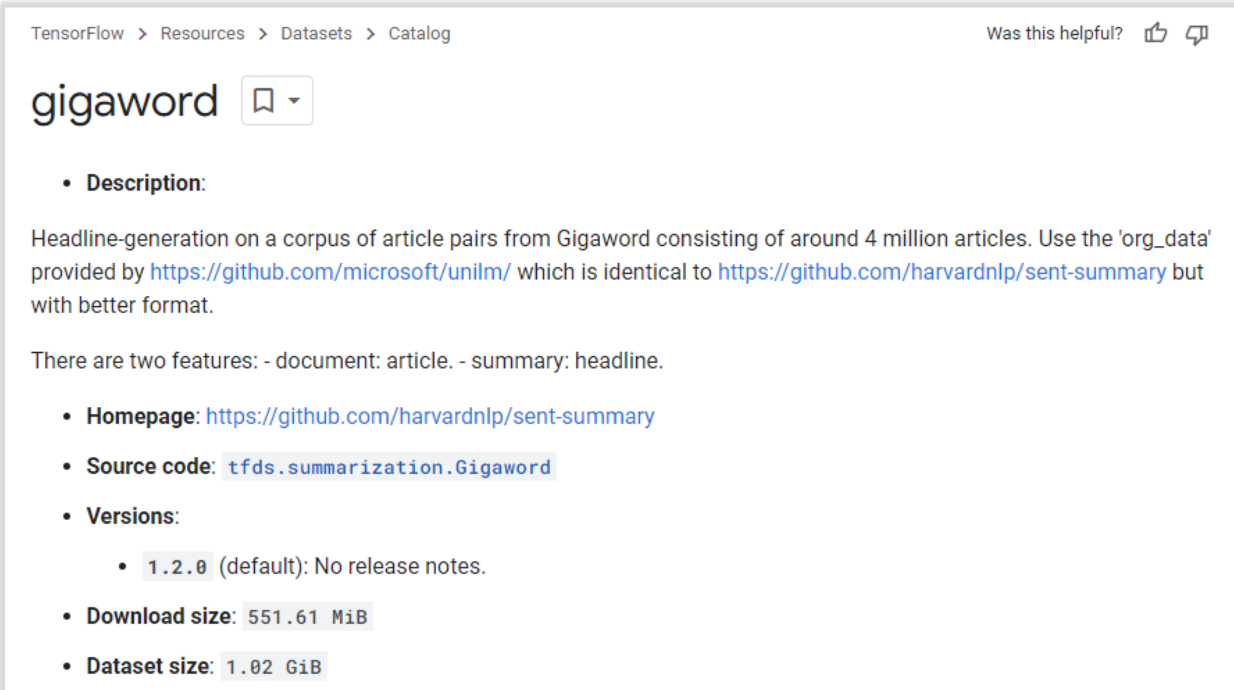
CNN/DailyMail non-anonymized summarization dataset.

There are two features: - article: text of news article, used as the document to be summarized - highlights: joined text of highlights with ~~and~~ around each highlight, which is the target summary

- **Additional Documentation**: Explore on Papers With Code ↗
- **Homepage**: https://github.com/abisee/cnn-dailymail
- **Source code**: `tfds.summarization.CnnDailymail`
- **Versions**:
  - `1.0.0`: New split API (https://tensorflow.org/datasets/splits)
  - `2.0.0`: Separate target sentences with newline. (Having the model predict newline separators makes it easier to evaluate using summary-level ROUGE.)
  - `3.0.0`: Using cased version.
  - `3.1.0`: Removed BuilderConfig
  - `3.2.0`: Remove extra space before added sentence period. This shouldn't affect ROUGE scores because punctuation is removed.
  - `3.3.0`: Add publisher feature.
  - `3.4.0` (default): Add ID feature.
- **Download size**: `558.32 MiB`
- **Dataset size**: `1.29 GiB`

- **Gigaword**

Figure 34: Gigaword Dataset (view)



TensorFlow > Resources > Datasets > Catalog

Was this helpful?

gigaword

- **Description**:

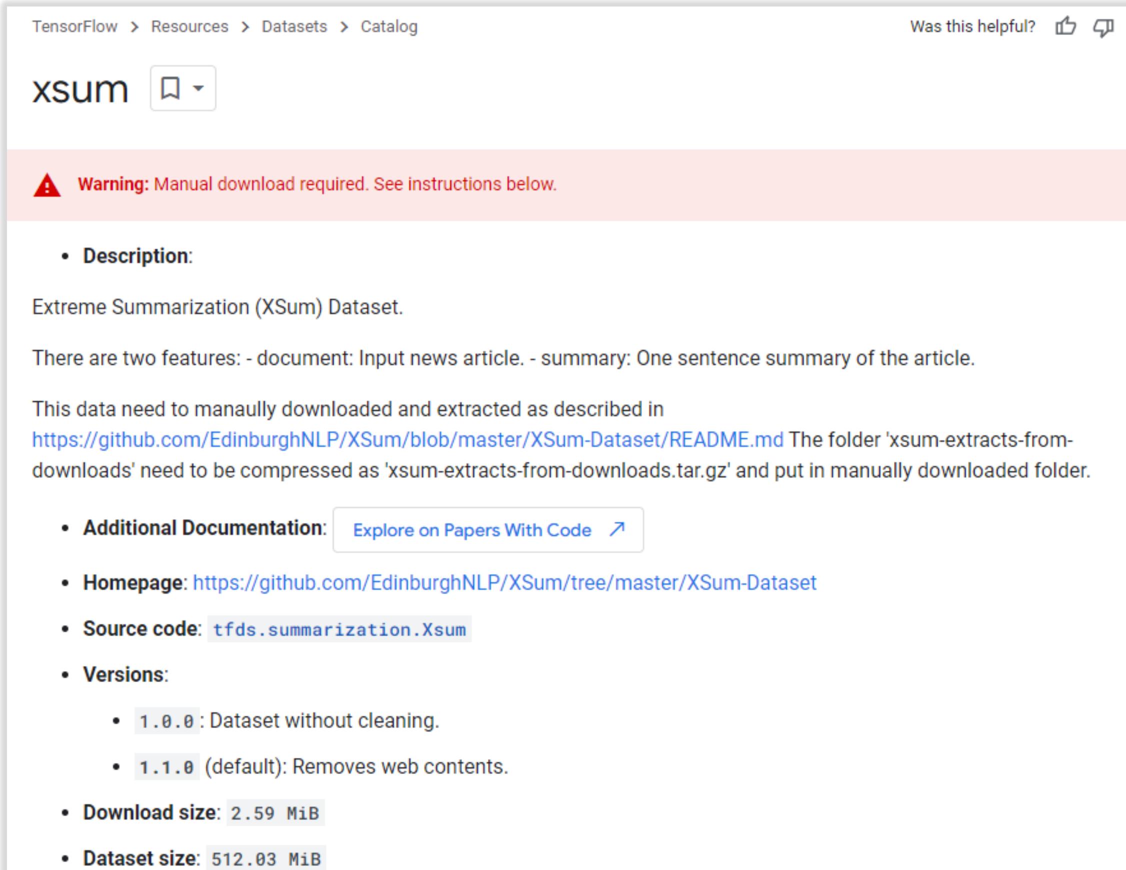
Headline-generation on a corpus of article pairs from Gigaword consisting of around 4 million articles. Use the 'org_data' provided by https://github.com/microsoft/unilm/ which is identical to https://github.com/harvardnlp/sent-summary but with better format.

There are two features: - document: article. - summary: headline.

- **Homepage**: https://github.com/harvardnlp/sent-summary
- **Source code**: `tfds.summarization.Gigaword`
- **Versions**:
  - `1.2.0` (default): No release notes.
- **Download size**: `551.61 MiB`
- **Dataset size**: `1.02 GiB`

- **Xsum**

Figure 35: Xsum Dataset (view)


TensorFlow > Resources > Datasets > Catalog Was this helpful?

xsum

Warning: Manual download required. See instructions below.

- **Description**:

Extreme Summarization (XSum) Dataset.

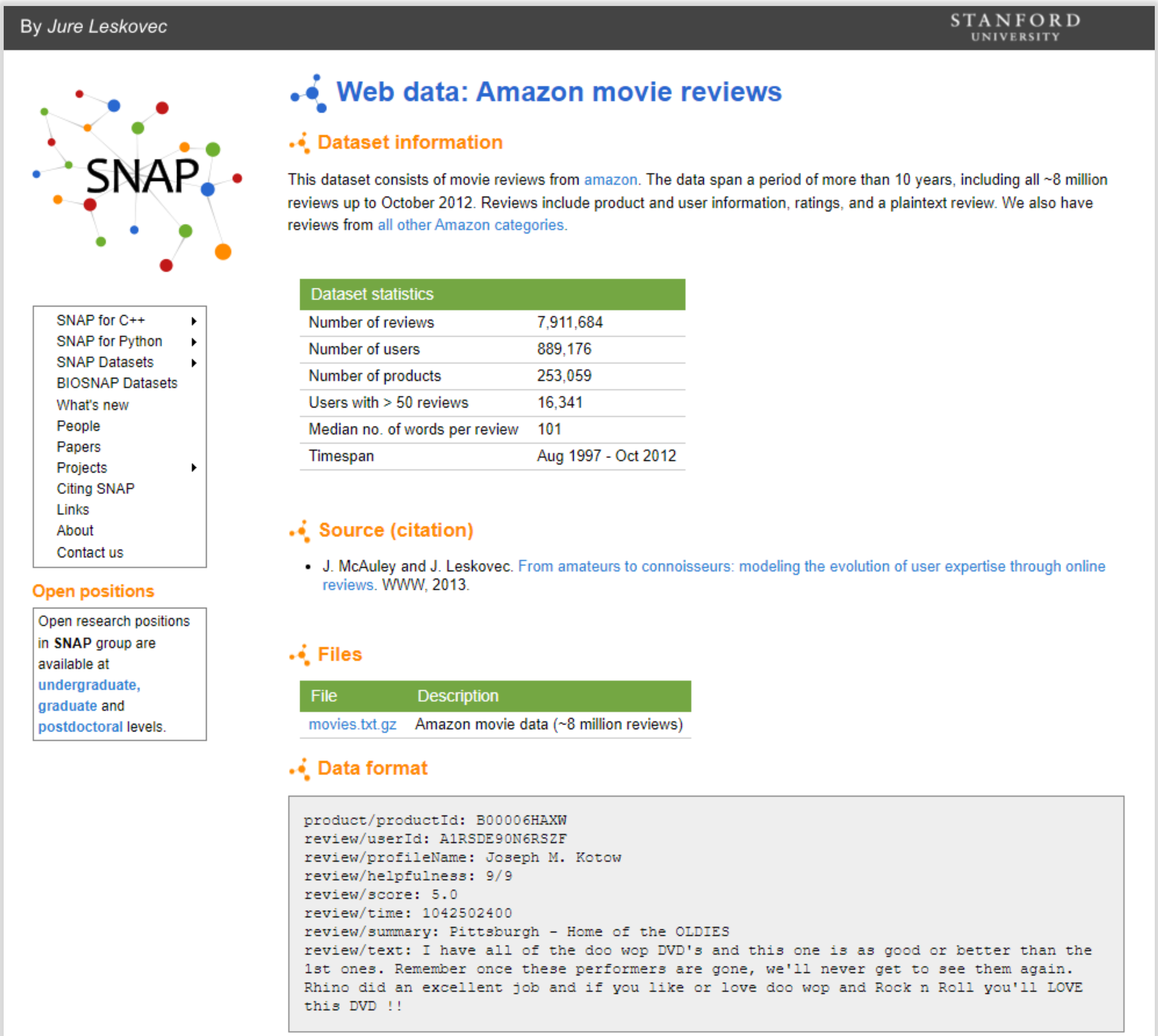
There are two features: - document: Input news article. - summary: One sentence summary of the article.

This data need to manaully downloaded and extracted as described in https://github.com/EdinburghNLP/XSum/blob/master/XSum-Dataset/README.md The folder 'xsum-extracts-from-downloads' need to be compressed as 'xsum-extracts-from-downloads.tar.gz' and put in manually downloaded folder.

- **Additional Documentation**: Explore on Papers With Code
- **Homepage**: https://github.com/EdinburghNLP/XSum/tree/master/XSum-Dataset
- **Source code**: `tfds.summarization.Xsum`
- **Versions**:
  - `1.0.0`: Dataset without cleaning.
  - `1.1.0` (default): Removes web contents.
- **Download size**: `2.59 MiB`
- **Dataset size**: `512.03 MiB`

- **Amazon movie reviews**

Figure 36: Amazon Movie Reviews Dataset (view)


By Jure Leskovec

STANFORD UNIVERSITY

SNAP

- SNAP for C++
- SNAP for Python
- SNAP Datasets
- BIOSNAP Datasets
- What's new
- People
- Papers
- Projects
- Citing SNAP
- Links
- About
- Contact us

Open positions

Open research positions in **SNAP** group are available at **undergraduate**, **graduate** and **postdoctoral** levels.

Web data: Amazon movie reviews

Dataset information

This dataset consists of movie reviews from amazon. The data span a period of more than 10 years, including all ~8 million reviews up to October 2012. Reviews include product and user information, ratings, and a plaintext review. We also have reviews from all other Amazon categories.

Dataset statistics	
Number of reviews	7,911,684
Number of users	889,176
Number of products	253,059
Users with > 50 reviews	16,341
Median no. of words per review	101
Timespan	Aug 1997 - Oct 2012

Source (citation)

- J. McAuley and J. Leskovec. From amateurs to connoisseurs: modeling the evolution of user expertise through online reviews. WWW, 2013.

Files

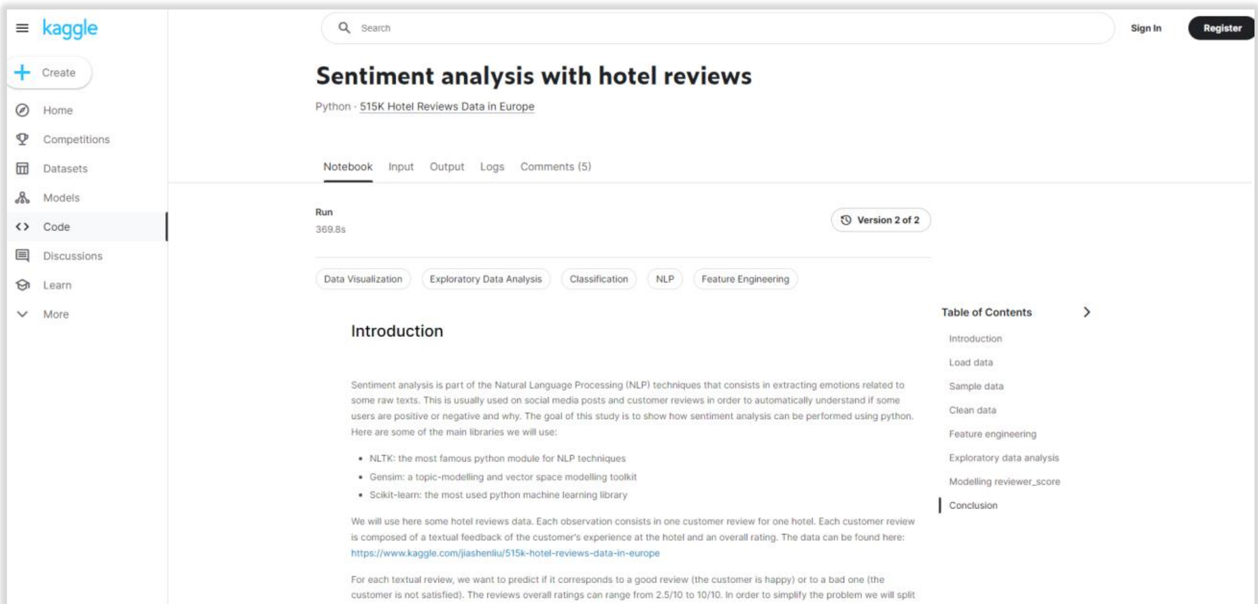
File	Description
movies.txt.gz	Amazon movie data (~8 million reviews)

Data format

```
product/productId: B00006HAXW
review/userId: A1RSDE90N6RSZF
review/profileName: Joseph M. Kotow
review/helpfulness: 9/9
review/score: 5.0
review/time: 1042502400
review/summary: Pittsburgh - Home of the OLDIES
review/text: I have all of the doo wop DVD's and this one is as good or better than the
1st ones. Remember once these performers are gone, we'll never get to see them again.
Rhino did an excellent job and if you like or love doo wop and Rock n Roll you'll LOVE
this DVD !!
```

- **Hotel Reviews**

Figure 37: Hotel Reviews Dataset (view)



E.2. Selection of Programming Language

The following table provides a summary of the evaluation of the programming language selected for the data science segment, in which each alternative was assigned a score ranging from H (High), M (Medium), to L (Low).

Table 53: Selection of Data Science Language

Data science An evaluation was conducted on two commonly utilized programming languages for data science to determine their suitability for implementing the fundamental components of data science.			
Aspect	**Relevance**	**Python**	**R**
Library availability and accessibility.	Essential to have a language that supports multiple libraries for the author to choose from, to gather data and build the model.	H	M

Author experience and implementation ease.	Efforts should be made to simplify the model application process, and it would be helpful if the author has experience with the chosen language.	H	M
Learning curve	Progress should not be hindered by the language's complexity since the goal is to utilize it as a tool to build a system, not to spend time learning it unnecessarily (Virtanen et al., 1970).	L	M
Documentation and Community	Clear documentation and a supportive community are vital since the author cannot afford to fix minor issues themselves.	H	M
Conclusion After analyzing the options, the author opted for **Python** since it was more suitable.			

E.3. Selection of Deep Learning (DL) framework

Table 54: Selection of DL framework

Framework	Description
TensorFlow	This tool is geared towards production-level applications and can handle large datasets. It is equipped with comprehensive documentation and has a supportive community. Additionally, it offers improved visualization options that simplify the process of debugging and monitoring training (Abadi et al., 1970).
PyTorch	This tool has a higher-level development, making it more lightweight and user-friendly. It has a smaller learning curve, making it easier to start using, and it feels more intuitive since building models is simpler (Paszke et al., 1970).
Conclusion The author opted to use **PyTorch** due to lightweight and user-friendly feature which makes it easier to work with (Paszke et al., 1970).	

E.4. Selection of User Interface (UI) framework

Table 55: Selection of UI framework

Framework	Description
Angular	This tool is appropriate for large-scale applications and includes dedicated submodules for specific functionalities. However, it may be less performant than other options and can be unnecessarily heavy (Waranashiwar & Ukey, 1970).
Vue	This framework is small and starts up quickly, and its code is straightforward, making it easy to use. Simulations have shown that it outperforms Angular and React. However, it has significantly fewer resources available (Wahyudi et al., 1970).
Svelte	This option is the most lightweight and reactive, offering superior performance compared to others. However, it has a small community of developers and is relatively new.
React	This option allows for customization and promotes code reusability through functions as components. It has a large and active community and is open-source, as well as being SEO-friendly. Furthermore, the React developer tools are very useful (Verma et al., 1970).
Conclusion After analyzing the options, the author selected **React** for building the GUI since it will be simple, and there is no need for a tool capable of handling large-scale applications, which is not the main focus (Verma et al., 1970).	

E.5. Selection of Application Programming Interface (API) framework

Table 56: Selection of web framework

Framework	Description
Flask	This framework is extremely lightweight and offers only basic functionality. Nonetheless, it is the preferred option for ML API development due to its lightness (Relan, 1970).

Django	This option is appropriate for larger-scale applications that require a wide range of functionalities. However, it is more rigid and less flexible, making it more demanding and heavier (Srivastava, 1970).
Conclusion The author opted for **Flask** since it provides only the essential features required for exposing an ML model.	

E.6. Data Preprocessing

Figure 38: Preprocessing: Remove Markdown (Self-Composed)

```
def md_links(text: Text) -> Text:
    markdown_link=re.compile(r'\[.*?\]\(.*?\)')
    return markdown_link.sub(r'',text)
```

```
df['text'] = df['text'].parallel_apply(lambda sentence: md_links(sentence))
df['summary'] = df['summary'].parallel_apply(lambda sentence: md_links(sentence))
```

Figure 39: Preprocessing – Remove Hyperlinks (Self-Composed)

```
def scrape_links(text):
    url = re.compile(r'https?://\S+|www\.\S+')
    return url.sub(r'',text)
```

```
df['text'] = df['text'].parallel_apply(lambda sentence: scrape_links(sentence))
df['summary'] = df['summary'].parallel_apply(lambda sentence: scrape_links(sentence))
```

Figure 40: Preprocessing: Remove Html Tags (Self-Composed)

```
def remove_html_tags(text: Text) -> Text:
    html=re.compile(r'<.*?>')
    return html.sub(r'',text)
```

```
df['text'] = df['text'].parallel_apply(lambda sentence: remove_html_tags(sentence))
df['summary'] = df['summary'].parallel_apply(lambda sentence: remove_html_tags(sentence))
```

Figure 41: Preprocessing: Char Words Extension (Self-Composed)

```
def chat_words_conversion(text: Text) -> Text:
    new_text = []
    for word in text.split():
        if word.upper() in chat_words_map_dict:
            new_text.append(chat_words_map_dict[word.upper()])
        else:
            new_text.append(word)
    return " ".join(new_text)
```

```
df['text'] = df['text'].parallel_apply(lambda sentence: chat_words_conversion(sentence))
df['summary'] = df['summary'].parallel_apply(lambda sentence: chat_words_conversion(sentence))
```

The above code snippets are used to convert the short key words into longer form, such as e.g.: ‘ATM’ is converted into ‘At the moment’

Figure 42: Preprocessing: Handling Common Contractions (Self-Composed)

```
def en_contractions(text: Text) -> Text:
    return ' '.join([contractions.fix(word)
                     if word in contractions.contractions_dict else word
                     for word in text.split()])
```

```
df['text'] = df['text'].parallel_apply(lambda sentence: en_contractions(sentence))
df['summary'] = df['summary'].parallel_apply(lambda sentence: en_contractions(sentence))
```

The above code snippets are used to handle/extend common contractions such as e.g.: ‘They’re’ into ‘They are’

Figure 43: Preprocessing: Removing Special Characters (Self-Composed)

```
s_chars = '¥₽ĬïŰŬĎŸæ₿₪ÚŇÀèÅ”ĜåŽÖéříÿý€ŝĤ₹áŜŮÂ≥ûÌÇšŘúüëÓđŠčÎŤÆÒœ₩öËäøÍťìĈôàĥÝ¢ç“žõÙÊĉŭÈŒÐÉÔĵùÁů„âÄűĴóêĝÞîØòď฿ČÜþňÛ'
PUNC = '+@«#_\-!$%%^&*¬()£<>?/\\|}\]\[{;\,~:\"\''
```

```
def special_char(text: Text) -> Text:
    # first, let's remove any unicode strings
    text = text.encode('ascii', 'ignore').decode()
    # remove printable bachslashes
    text = re.sub(r'[\t\s\n\r\b\a]', ' ', text)
    # Special letters
    text = re.sub(r'[{}]'.format(s_chars), '', text)
    # Punctuation [remove punctuation between spaces only which represent noises]
    text = re.sub(r'\s[{}]\s'.format(PUNC), ' ', text)
    # space at the start or the end of the context
    text = re.sub(r'(^\s)|(\s$)', '', text)
    # Single character
    text = re.sub(r'(\s[^iIaA]\s)', ' ', text)
    return text
```

```
df['text'] = df['text'].parallel_apply(lambda sentence: special_char(sentence))
df['summary'] = df['summary'].parallel_apply(lambda sentence: special_char(sentence))
```

```
df.head(3)
```

Figure 44: Preprocessing: Resolving Spelling Mistakes (Self-Composed)

```
from textblob import TextBlob

def spell_correction(df):
    # creating a new column for the corrected text
    df['corrected_text'] = df['text']
    # creating a new column for the corrected summary
    df['corrected_summary'] = df['summary']
    # creating a for loop for the entire dataset
    for i in range(len(df)):
        # Records
        print('Counter: ' + str(i+1) + '/' + str(len(df)+1))
        # creating a variable for the text of the current row
        text = df['corrected_text'][i]
        # creating a variable for the summary of the current row
        summary = df['corrected_summary'][i]
        # creating a variable for the corrected text of the current row
        corrected_text = TextBlob(text).correct()
        # creating a variable for the corrected summary of the current row
        corrected_summary = TextBlob(summary).correct()
        # updating the corrected text column with the corrected text
        df['corrected_text'][i] = str(corrected_text)
        # updating the corrected summary column with the corrected summary
        df['corrected_summary'][i] = str(corrected_summary)
    # returning the dataset with the new columns
    return df

spell_correction(df)
df_copy_correction = df.copy()
```

Figure 45: Preprocessing: Removing Duplicates (Self-Composed)

```
def rm_duplicates(text: Text) -> Text:
    return re.sub(r'\b(\w+\s*)\1{1,}', '\\1', text)
```

```
df_copy_correction['corrected_text'] = df_copy_correction['corrected_text'].parallel_apply(lambda sentence: rm_duplicates(sentence))
df_copy_correction['corrected_summary'] = df_copy_correction['corrected_summary'].parallel_apply(lambda sentence: rm_duplicates(sentence))
```

Figure 46: Preprocessing: Restoring Missing Punctuations (Self-Composed)

```
args = InferenceArguments(
            model_name_or_path="Qishuai/distilbert_punctuator_en",
            tokenizer_name="Qishuai/distilbert_punctuator_en",
            tag2punctuator=DEFAULT_ENGLISH_TAG_PUNCTUATOR_MAP,
)
inference = Inference(inference_args=args, verbose=False)
```

```
def punct_restoration(list_of_text: List[Text], name: Text) -> List[Text]:
    list_of_texts = []
    for text in tqdm(list_of_text, desc=f"Auto Punctuation for {name}"):
        list_of_texts.append(
            inference.punctuation([text])[0][0]
        )
    return list_of_texts
```

```
df_copy_correction['punc_corrected_text'] = punct_restoration(df_copy_correction['corrected_text'].values.tolist(), "text")
df_copy_correction['punc_corrected_summary'] = punct_restoration(df_copy_correction['corrected_summary'].values.tolist(), 'summary')
```

Figure 47: Preprocessing: Grammarly Correction (Self-Composed)

```
def grammely_correction(list_text: List[Text], name: Text) -> List[Text]:
    list_of_correction = []
    for text in tqdm(list_text, desc=f'Grammerly Correction for {name}'):
        if len(text.split()) < 50:
            list_of_correction.append(list(gf.correct(text, max_candidates=1))[0])
        else:
            list_of_correction.append(" ".join([list(gf.correct(sentence, max_candidates=1))[0]
                                                for sentence in tokenizer.tokenize(text)]))
    return list_of_correction
```

```
df_copy_correction['gram_corrected_text'] = grammely_correction(df_copy_correction['punc_corrected_text'].values.tolist(), "text")
df_copy_correction['gram_corrected_summary'] = grammely_correction(df_copy_correction['punc_corrected_summary'].values.tolist(), 'summary')
```

E.7. User interface

Figure 48: GUI – Homepage (After Signup) (Self-Composed)

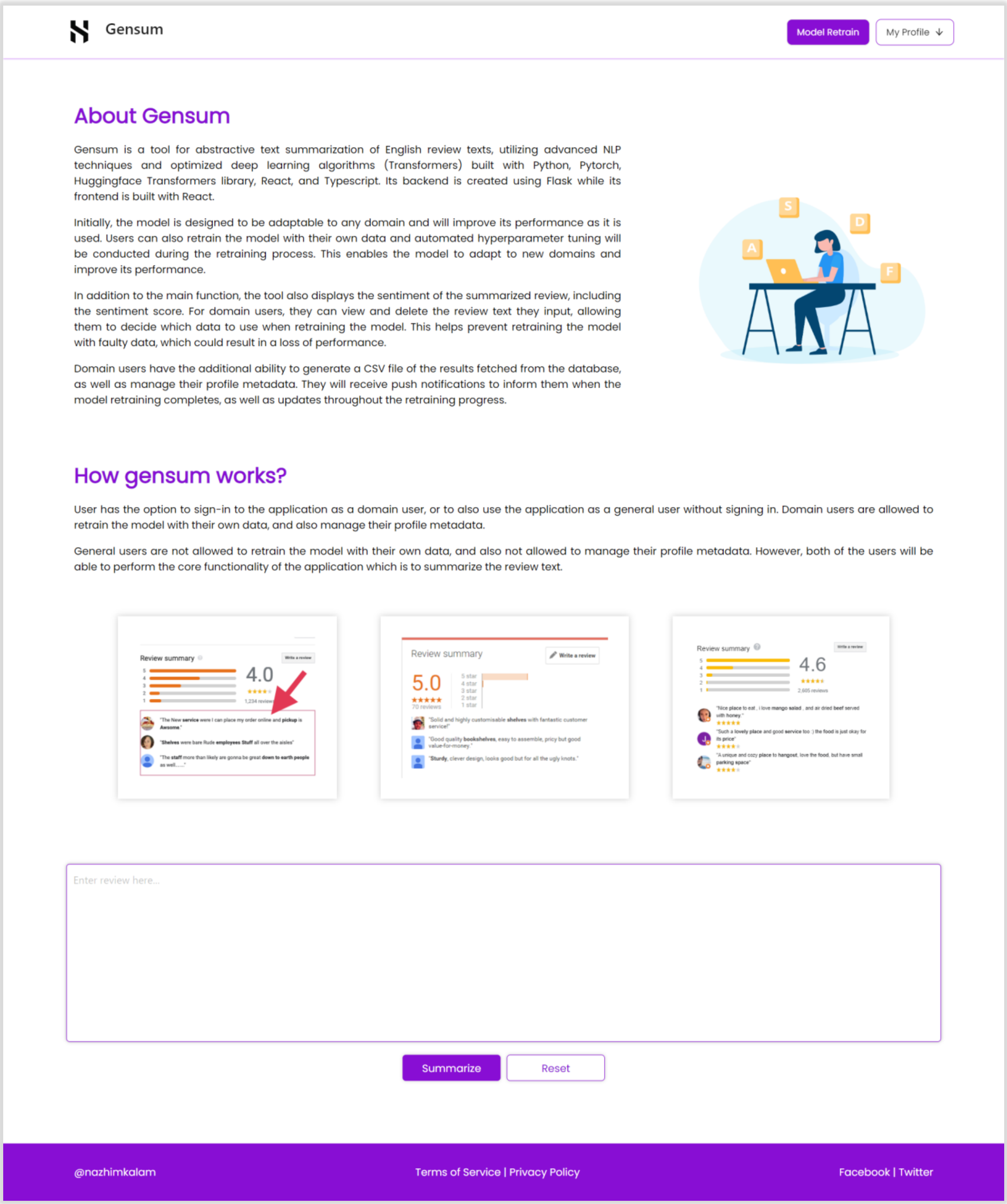


Figure 49: GUI – Review History Page (Self-Composed)

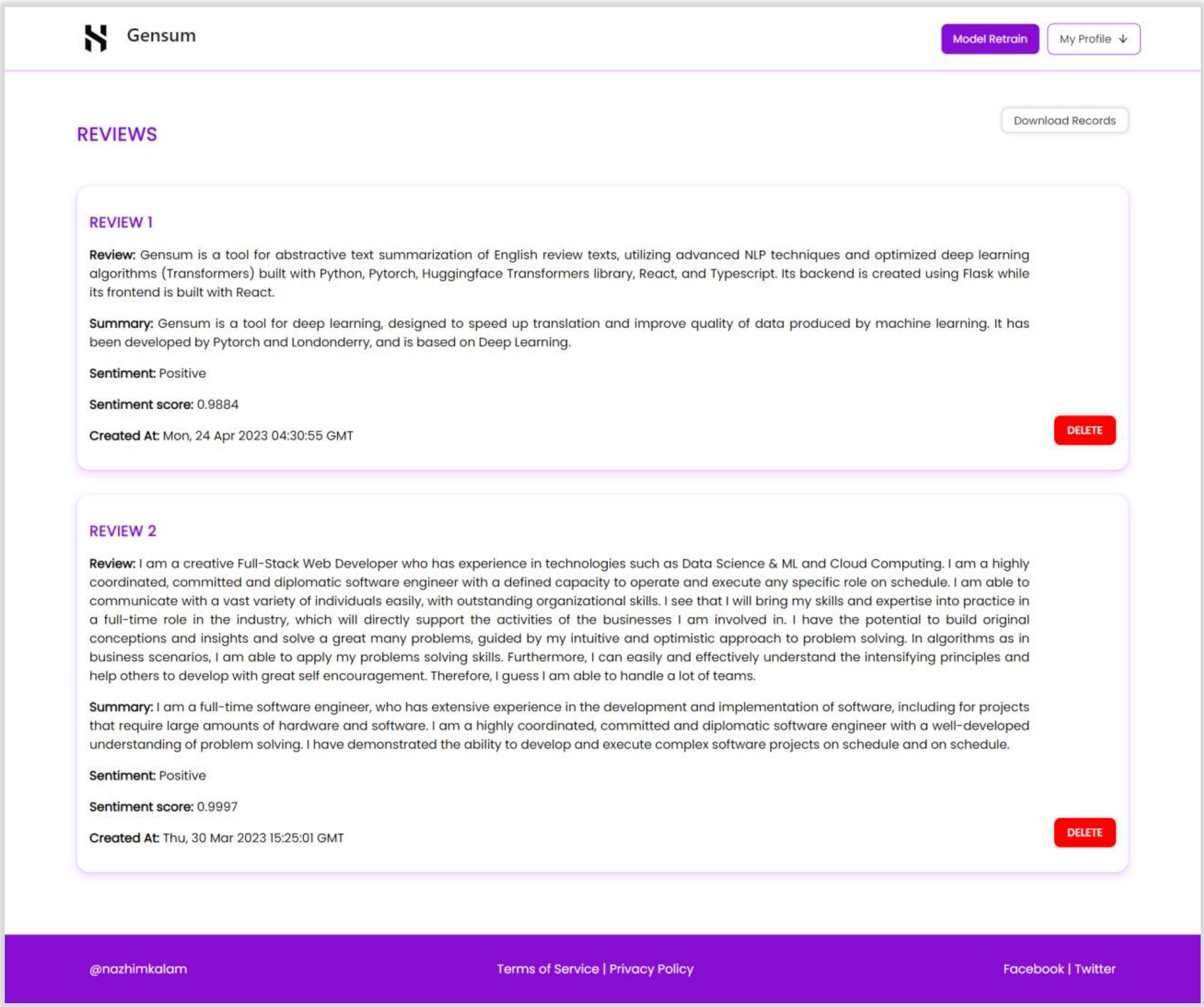


Figure 50: GUI – User Profile (Self-Composed)

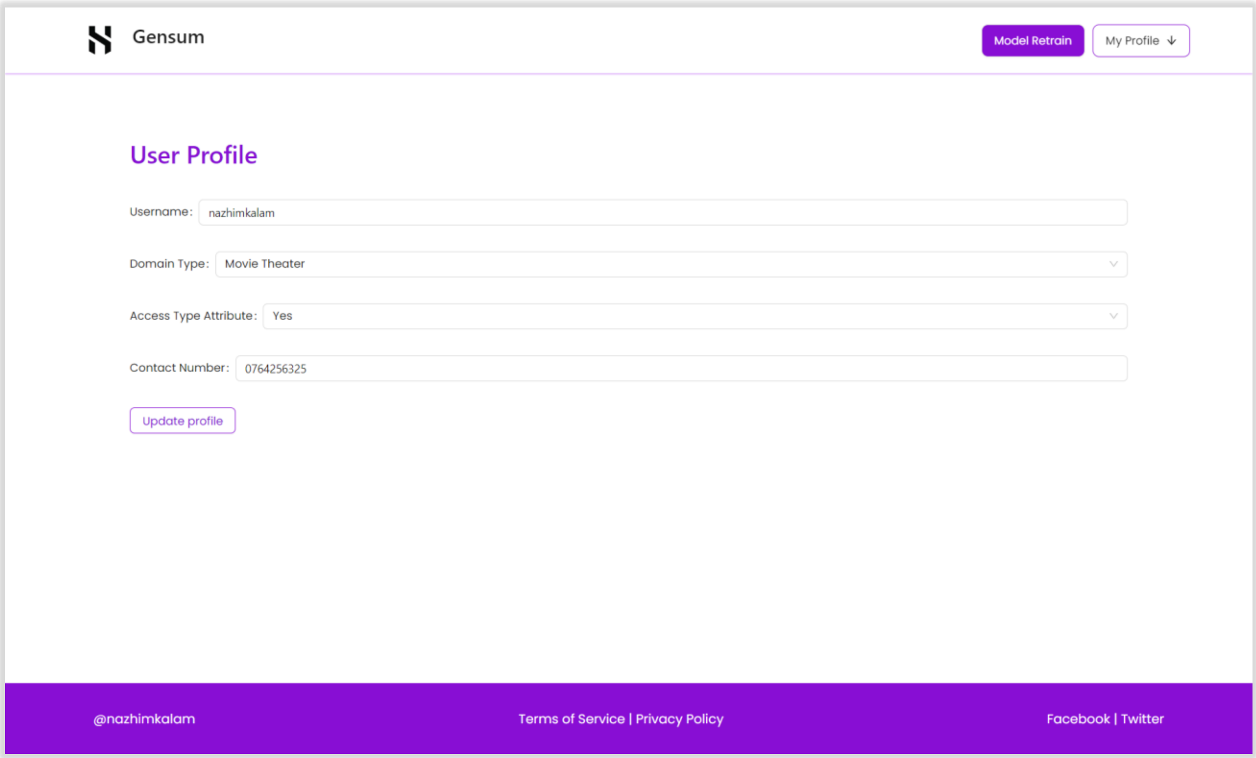


Figure 51: GUI – Model Retraining Modal (Self-Composed)

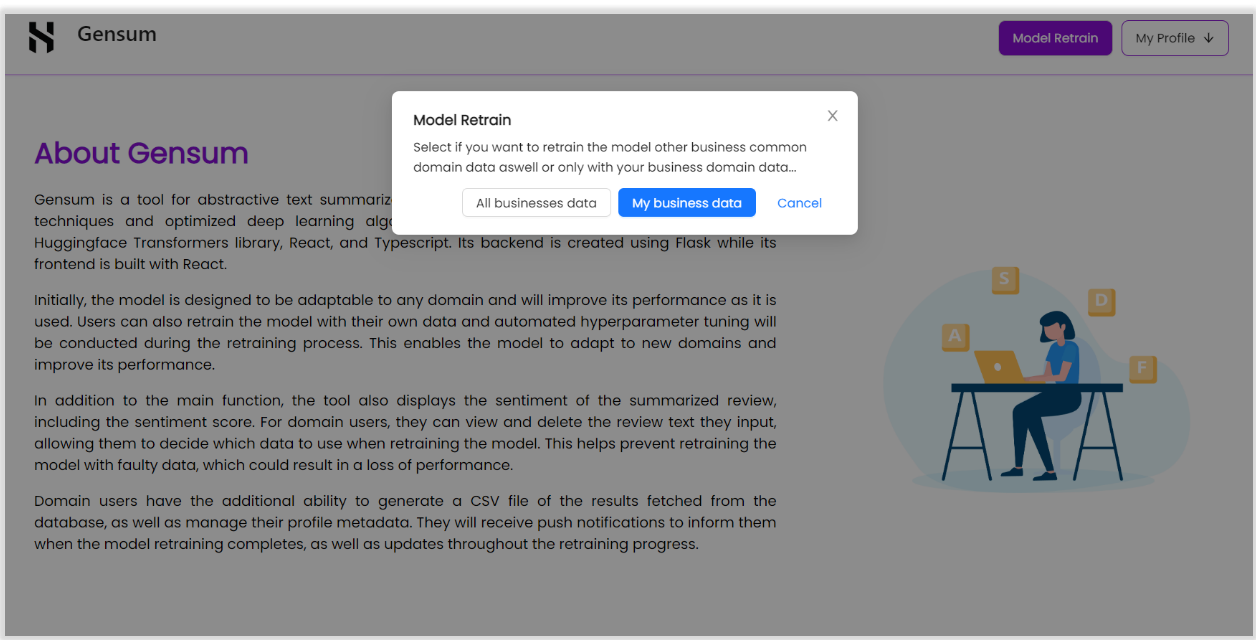
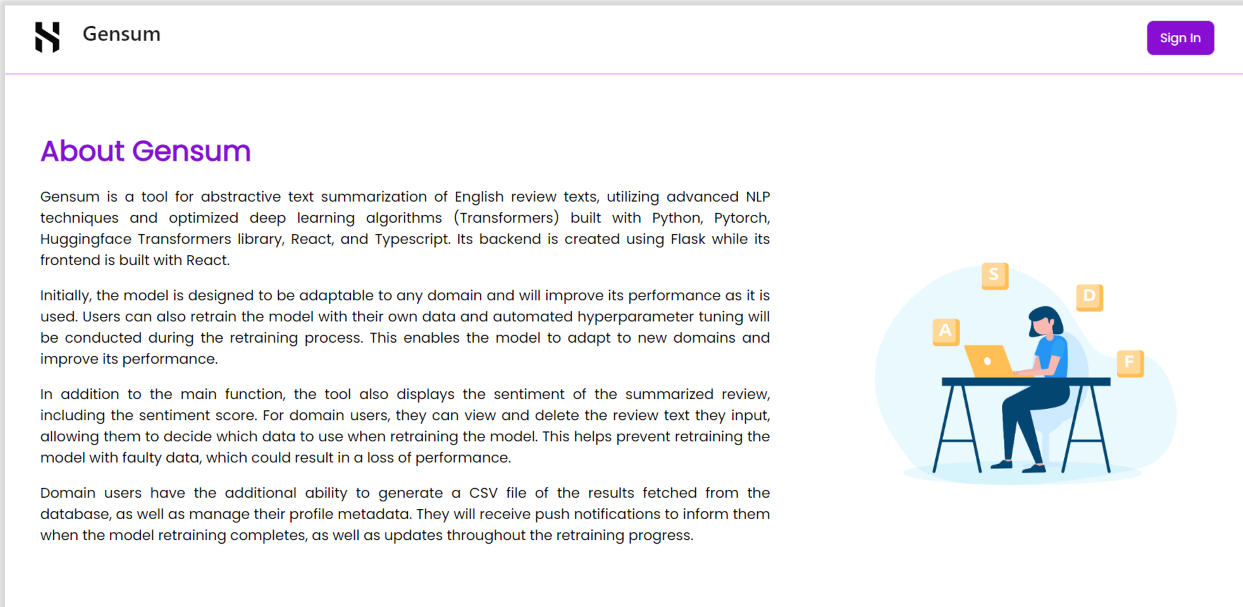


Figure 52: GUI – Homepage (Before Signup) (Self-Composed)



APPENDIX F – TESTING

F.1. Functional Testing

Table 57: Functional Testing

Test case	FR ID	User Action	Expected Result	Actual Result	Result Status
1	FR1	Users, both general and specific to the domain, can input review text to produce a summary.	Users can input review text in a provided field and generate a summary by clicking a button.	Users can input review text in a provided field and generate a summary by clicking a button.	Passed
2	FR2, FR3	Users will click "signup" to create an account, then go to their profile to update it.	After signing up, domain users will see their personal profile and can access their pages.	After signing up, domain users will see their personal profile and can access their pages.	Passed
3	FR4	If there is enough data in the database, domain users can click the "retrain model" button to initiate model retraining.	Users will receive a prompt to confirm model retraining, and email notifications will be sent to them to update them on the progress.	Users will receive a prompt to confirm model retraining, and email notifications will be sent to them to update them on the progress.	Passed
4	FR5	User will be able to trigger model retraining during off peak hours.	Model retraining happens	Model retraining happens	Passed

5	FR6, FR7, FR8, FR9	None	The system will find the new best set of hyperparameters by fetching the domain user data/groups and perform necessary preprocessing and trigger model retraining.	The system will find the new best set of hyperparameters by fetching the domain user data/groups and perform necessary preprocessing and trigger model retraining.	Passed
6	FR10	None	The summary output will be displayed on the UI	The summary output will be displayed on the UI	Passed
7	FR11	None	The system uses the latest retrained model for summary generation	The system uses the latest retrained model for summary generation	Passed
8	FR12	Users can input review text to generate a summary and receive the sentiment and sentiment score of the review.	Displays the summary text along with the sentiment score and the sentiment.	Displays the summary text along with the sentiment score and the sentiment.	Passed
9	FR14	None	The system encrypts the data and stores it in the database for model retraining purposes	The system encrypts the data and stores it in the database for model retraining purposes	Passed

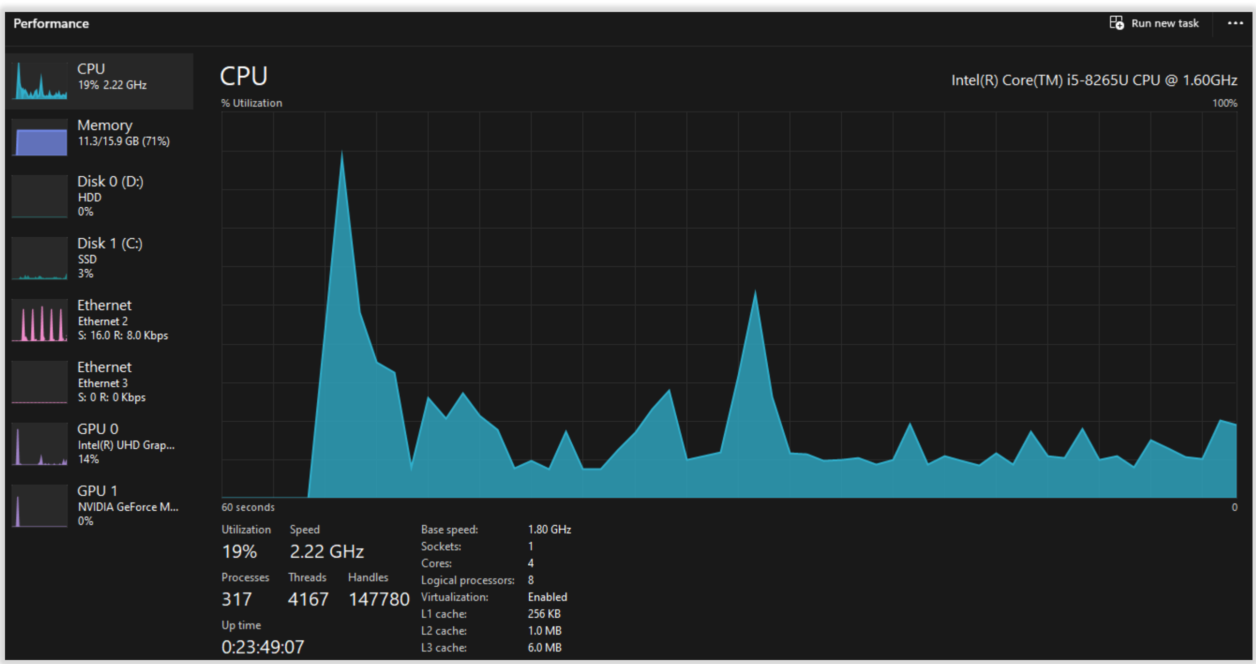
F.2. Non-functional testing

In order to evaluate whether the system satisfies the non-functional requirements and design objectives, the writer utilized testing methodologies such as performance testing, GUI testing, maintainability testing, and a limited number of test cases.

Performance testing

At present, the author has configured the system to operate in a local environment, and the system resource manager graph below depicts the utilization of resources when the application is deployed and operational in this environment.

Figure 53: System Resource Manager Graph (Self-Composed)



GUI testing

During the requirement gathering phase, it was established that developing a straightforward and efficient GUI was crucial. To assess its performance and accessibility, the GUI was evaluated using Google Lighthouse. The results of this evaluation are illustrated in the diagram below.

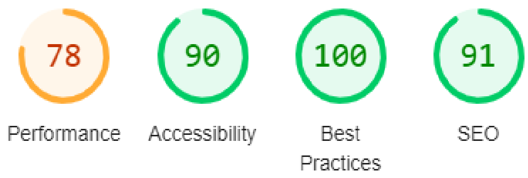


Figure 54: Lighthouse Landing Page

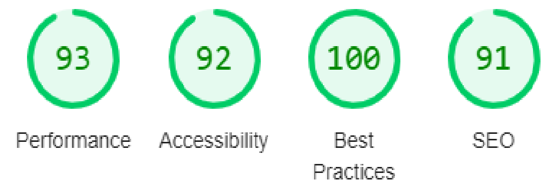


Figure 55: Lighthouse Records Page

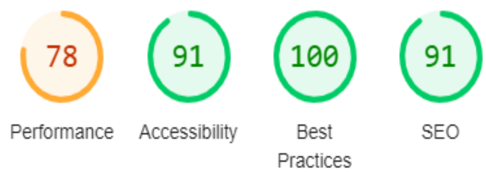


Figure 56: Lighthouse User Profile Page

Maintainability testing

Ensuring maintainability is essential to enable smooth future research on the system, particularly the developed algorithm. To achieve this, **CodeFactor** and **CodeQL** were employed to verify that the repositories are well-documented and maintained, and that there are no vulnerabilities present.

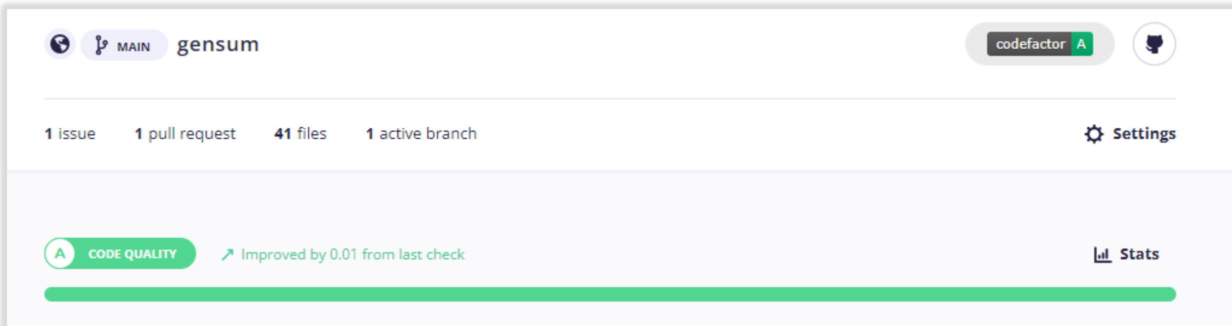


Figure 57: CodeFactor - Gensum Repository

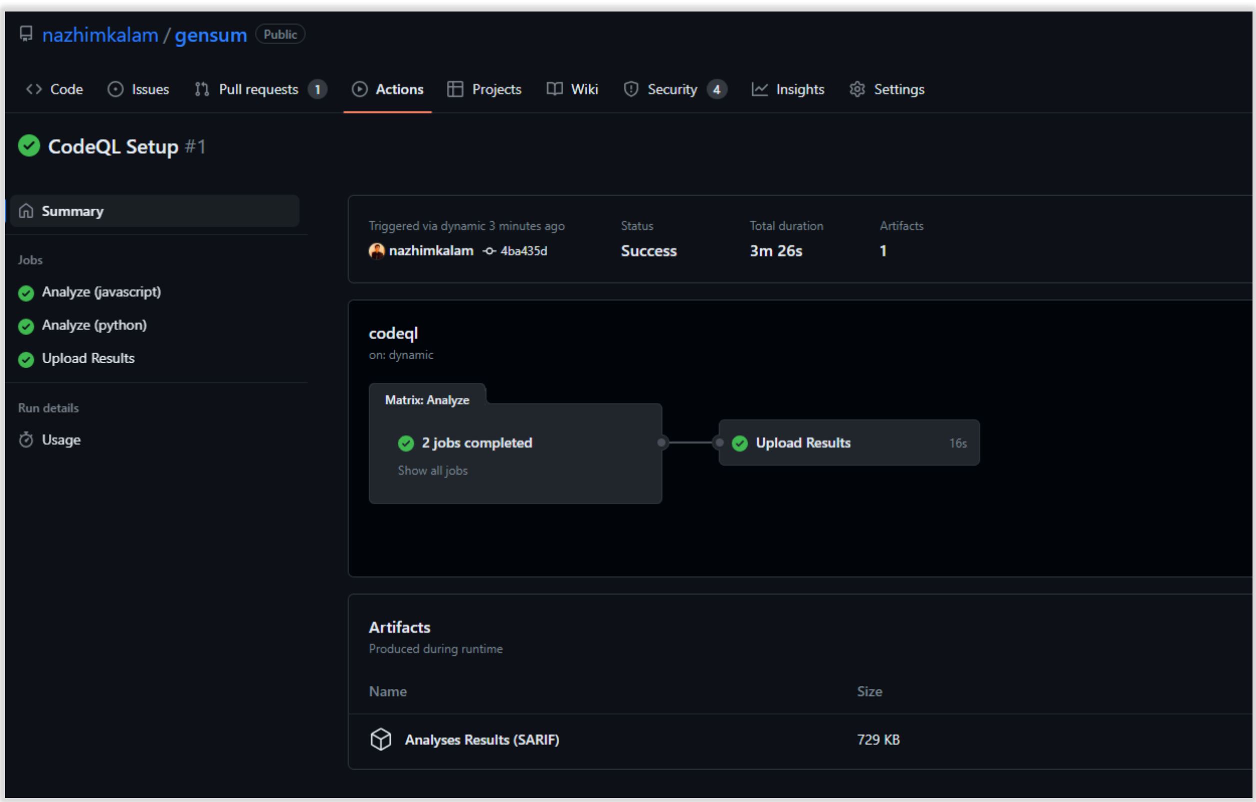


Figure 58: CodeQL - Gensum Repository

Test cases

Table 58: Non-functional testing

Non-functional requirements		
Test case	**ID**	**Result**
1	NFR1	Created a minimalist UI design making it user friendly to all type of users.
2	NFR2	Meaningful error messages are displayed on the GUI, if anything faulty happens.
3	NFR3	The summary generation on the GUI takes an average of 5000ms
4	NFR4	CodeQL and CodeFactor is used to maintain the coding standards up to the best quality.
5	NFR5	The idea of model generalization with respect on the domain users works as expected. A domain specific model is created whenever a new domain user signup into the application.
6	NFR6	Data encryption is applied for the data which are domain specific and stored in the database, to ensure that the meaning of the data is not retrained even if the data is lost.

APPENDIX G – EVALUATION

G.1. Expert evaluators

Table 59: Details of the expert evaluator(s) selected.

ID	Affiliation	Expertise related to the research
EV1	PhD Research Student in Computational Linguistics	NLP .
EV2	Machine Learning Expertise | Lecturer with PhD	ML and Neural Networks
EV3	NLP Researcher	NLP
EV4	Software Architect	Algorithms
EV5	Software Architect & ML Researchers	ML & Algorithms
EV6	VP Innovations, Software Engineer	ML & Neural Networks
EV7	Lecturer with MSc	NLP

G.2. Evaluation of functional requirements

Table 60: Evaluation of the implementation of functional requirements

ID	Description	Priority	Use Case	Evaluation
FR1	Both general and domain specific users must be able to enter a review text from the GUI considering as the starting point of the summary generation.	M	UC01	Implemented
FR2	Only Domain Specific Users should be able to sign up and create an account after entering the necessary details required	S	UC02	Implemented
FR3	The system could allow the ability to update the account details of the domain user after creating the account	C	UC02	Implemented

FR4	The system must undergo model retraining with the new data stored in the database for the specific domain user, when its triggered from the GUI with the user's consent	M	UC03	Implemented
FR5	The system could be able to perform model retraining automatically during off peak hours every day.	C	UC03	Implemented
FR6	The system must be able to find the new set of best hyperparameters with the usage of the new data.	M	UC04	Implemented
FR7	The system must be able to able to retrain the model with the new best hyperparameters and create the model	M	UC05	Implemented
FR8	The system must be able to pull the new data from the database to recreate the new dataset for retraining.	M	UC06	Implemented
FR9	The system should be able to combine all the data from a common group of domains when creating the dataset only given that the consent is approved to use their data	C	UC06	Implemented
FR10	The system must be able to process the review text and display the summary output on the GUI	M	UC07	Implemented
FR11	The system must be able to use the latest trained model to generate the summary for the review text	M	UC08	Implemented
FR12	The system could also find the sentiment of the generated summary if its positive or negative and return the result.	C	UC08	Implemented
FR13	The system could make use of a hybrid model for the text summarization.	C	N/A	Not-considered
FR14	The system must store the entered user review and generated summary to be stored in the database for retraining purposes.	M	UC09	Implemented

FR15	The system should encrypt the data when saving into the database (both the review and summary)	S	UC09	Implemented
FR16	The system could allow the domain users to delete the reviews from the database.	C	UC10	Implemented
The percentage of functional requirements that have been fulfilled = $\frac{15}{16} * 100 = 93.75\%$				

G.3. Evaluation of non-functional requirements

Table 61: Evaluation of the implementation of non-functional requirements

ID	Specification	Description	Priority	Evaluation
NFR1	Usability	The system needs to be simple enough for non-technical individuals to utilize without much effort.	Important	Implemented
NFR2	Usability	Meaningful error messages should be displayed if anything goes wrong	Desirable	Implemented
NFR3	Performance	Summary generation should be done within 3000ms	Important	Implemented
NFR4	Maintainability	Following coding standards and best practices	Important	Implemented
NFR5	Generalization	Any domain users are able to use the application and model performance will adapt with respect to the domain	Important	Implemented
NFR6	Security	The system should protect against data corruption by attackers, and testing can ensure this.	Desirable	Implemented
NFR7	Scalability	The prototype can be used by several domains and multiple businesses under a single domain, then the system may have to support many concurrent user-requests.	Desirable	Not - Implemented Fully – (Database is set to be scaled)

The percentage of non-functional requirements that have been fulfilled = $\frac{6.5}{7} * 100 = 92.9\%$

Table 62: Evaluation of the achievement of design goals

ID	Goal	Evaluation
DG1	Performance	Achieved
DG2	Usability	Achieved
DG3	Quality	Achieved
DG4	Maintainability	Achieved
The percentage of achievement of design objectives. $\frac{4}{4} * 100 = 100\%$		

APPENDIX H – CONCLUSION

H.1. Status of Research Objectives

Table 63: Status of research objectives

Objective	Description	Status
Problem Identification	Comprehend and document the identified issue. **RO1:** Perform research in a domain of interest and identify a sufficiently comprehensive problem that needs to be addressed. **RO2**: Thoroughly explore and analyze potential solutions for addressing the problem. **RO3**: Explore methods for developing an adaptive and generalized approach. **RO4**: Create a schedule, determine associated deliverables, and develop a Gantt chart for the project.	Completed
Literature Review	Complete a thorough critical review of earlier related work. **RO1:** Make a preliminary investigation on existing abstractive text summarization using deep learning approaches. **RO2:** Make a preliminary investigation on why transformers architecture was the chosen deep learning choice for this research. **RO3:** Analyze the top tier transformer architectures widely used. **RO4:** Analyzing how the models can be fine-tuned via hyperparameter optimization. **RO5:** Analyzing the different approaches used for model evaluation. **RO6:** Analyze how the model can be generalized for every other domain.	Completed

Requirement Elicitation	Defining the project's needs utilizing relevant approaches and tools in order to solve the projected research gaps and obstacles based on prior related research. **RO1:** Gathering information related to the expected metadata required for the dataset to contain for the model training. **RO2:** Gathering the requirements of transformer architectures for fine-tuning and understand the end to end user expectations. **RO3:** Getting insights from domain experts to build a suitable system. **RO4:** Gathering the requirements for handling generalization.	Completed
Design	Considering the following when developing the suggested system: **RO1:** Design a component to preprocess the dataset for the respective model inputs. **RO2:** Design a component to store the top tier transformer models with their respective metadata, to use throughout. **RO3:** Design a hyperparameter tuning component that can improve accuracy of the transformer model. **RO4**: Design high-level architecture for the system.	Completed
Implementation	Setting up a mechanism capable of addressing the gaps that were intended to be covered. **RO1**: To develop data preprocessing component. **RO2**: To develop a component that handles and stores the top tier transformer architectures for fine-tuning. **RO3**: To develop the automated hyperparameter search component that handles all the top tier architectures assigned. **RO4**: To develop a component for the model evaluations for the measured hyperparameters **RO5**: Consider legal, social, ethical, and professional issues during implementation.	Completed

Evaluation	Effectively test the algorithm implemented, the system, and the respective data science model using recommended techniques. **R**O27: Evaluate the developed algorithm and the respective model against the evaluation metrics researched in the literature review. **R**O28: Create a test plan & test cases and perform unit, performance, and integration testing.	Completed
Documentation	Record the progress of the research project and report any encountered challenges. **R**O29: Produce a cohesive report that details newly acquired skills, evaluations, contributions, etc., and verify that all previously stated objectives have been accomplished.	Completed

H.2. Achievement of Learning Outcomes

Table 64: Achievement of learning outcomes

Description	**LO(s)**
The project was divided into smaller problems, and each problem was further broken down into manageable parts. Each part was addressed separately by using relevant techniques obtained from analyzing literature and project requirements.	LO1
The project plan was designed by identifying the components and setting them as milestones to be accomplished within a specific timeframe, ensuring the project's timely completion.	LO2
The author collected and analyzed project requirements from two groups: academic researchers and end-users, which provided valuable insights for developing the system and determining which features to prioritize.	LO3
The author critically reviewed the literature to comprehend the concepts pertinent to the domain.	LO4
Upon obtaining the requirements, insights, and knowledge, the author proceeded to address the two subproblems incrementally, acquiring new skills where necessary. The supervisor provided regular guidance to ensure that the project was progressing	LO5, LO6, LO7

as planned, with milestone deliverables being produced accordingly. Any SLEP considerations were considered and documented as well.	
The author documented the progress of the research by presenting each chapter to the supervisor as a milestone and incorporating feedback from both the supervisor and module leader. Prior to completing the dissertation, two document artifacts - the project proposal and PSDP - were submitted. Additionally, the author presented papers at conferences to validate their proposed solution.	LO8

H.3. Project Plan

Figure 59: Gantt Chart: Initial Plan (Self-Composed)

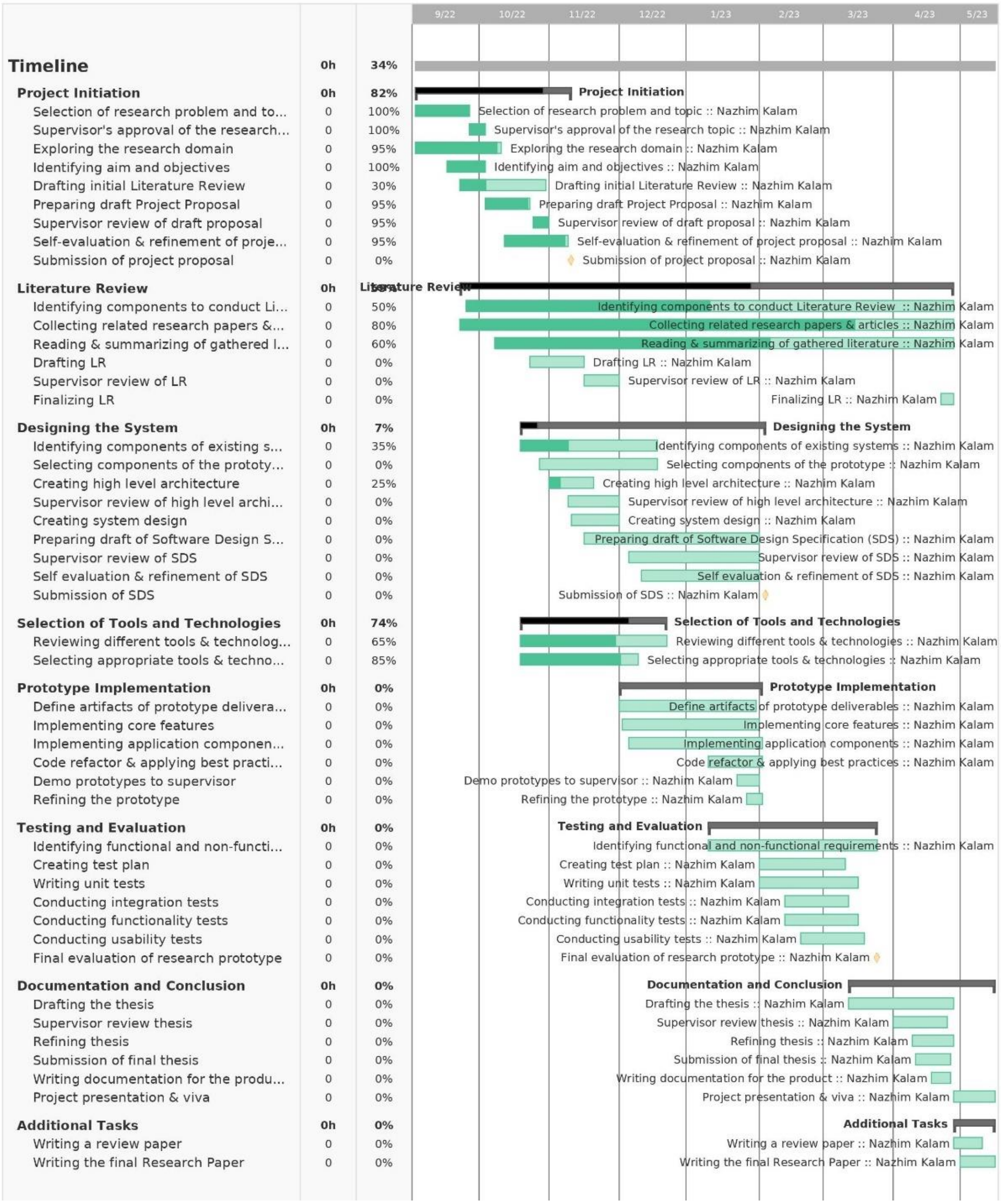
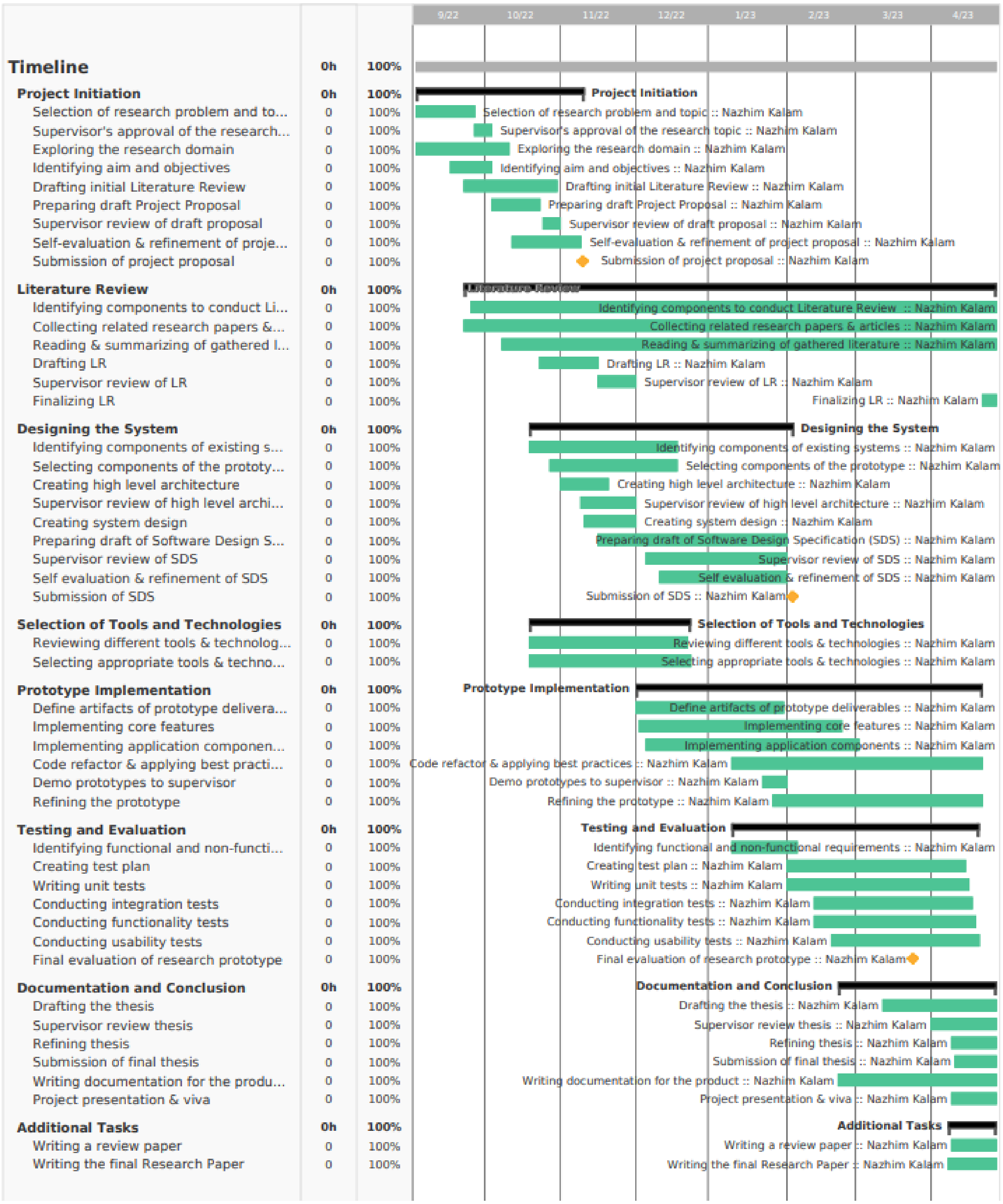
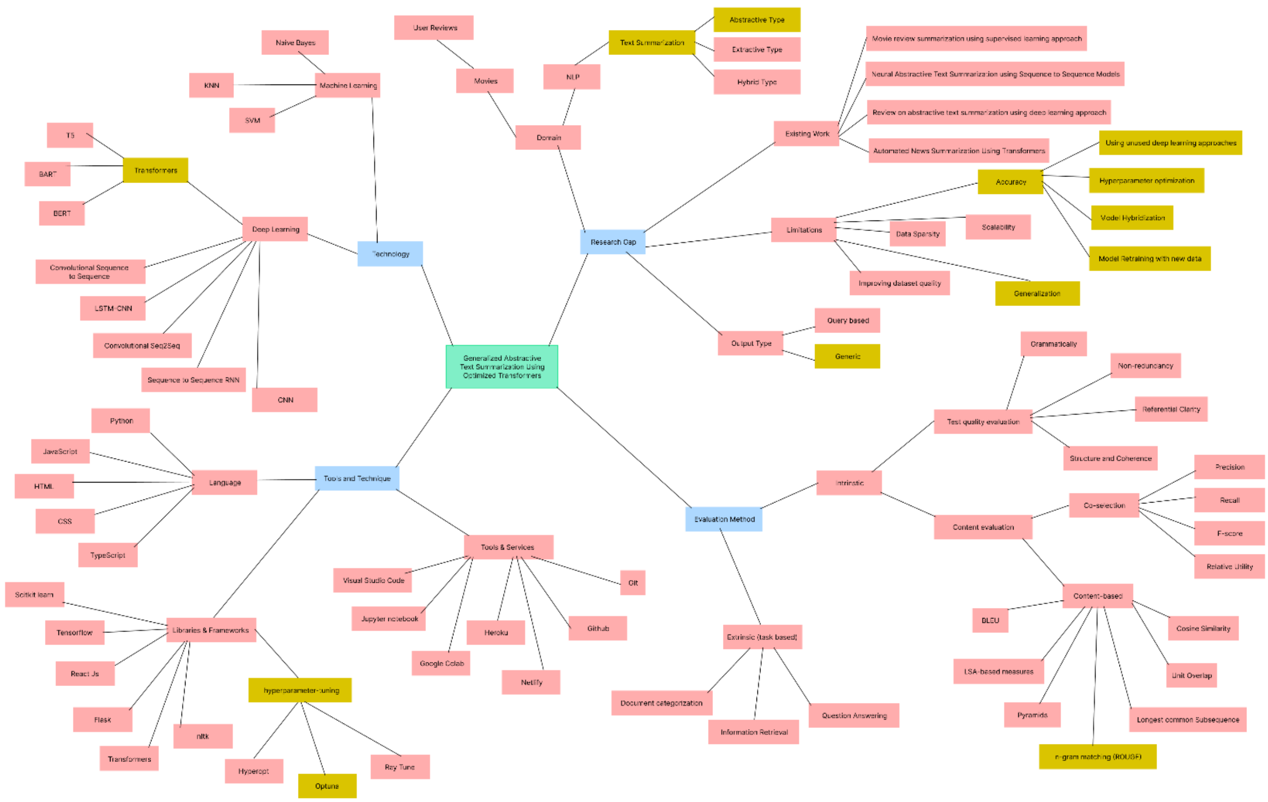


Figure 60: Gantt Chart: Final Plan (Self-Composed)



APPENDIX I – CONCEPT MAP



An enhanced version of the concept map can be found here.

APPENDIX J – REVIEW PAPER

A Review on Creating a Performance Adaptive Generalized Abstractive text Summarization using Optimized Transformers

NAZHIM KALAM, University of Westminster, UK
TORIN WIRASINGHA, Informatics Institute of Technology, Sri Lanka

Abstractive text summarization systems have been integrated with various applications in the world to perform text summarization, and it's nothing new to the field. However, with prior research it found that in the domain of movies the need for performance improvement is required using the latest approaches than the current traditional ML DL methods, movie review summarization plays a major role in helping users to make better decisions by matching their interest with the reviews of the movie, this saves a lot of time and also improves businesses in their sales.

In 2017 researchers from Google Brain introduced NLP Transformers, which is the latest approach to solving NLP problems, and it's increasingly been known and used nowadays over traditional ML DL approaches like using basic LSTM, and RNN approaches. The author explored ways in which to get an optimal solution using Transformer for abstractive text summarization and yet making a generalized solution that can be adapted with respect to any domain (be it hotels, movies, restaurants) and increase its performance as the system gets used over time.

This paper is a review of the approach taken to explore and construct a performance-adaptive generalized abstractive text summarizer using optimized transformers. The approach taken to optimize transformers will be discussed in this review.

CCS Concepts: • **Computing methodologies**; • **Artificial intelligence**; • **Natural language processing**; • **Natural language generation**;

Additional Key Words and Phrases: Natural Language Processing (NLP), Machine Learning (ML), Deep Learning (DL), Recall-Oriented Understudy for Gisting Evaluation (ROUGE), Inductive logic programming (ILP)

ACM Reference Format:
Nazhim Kalam and Torin Wirasingha. 2022. A Review on Creating a Performance Adaptive Generalized Abstractive text Summarization using Optimized Transformers. 1, 1 (April 2022), 8 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

A growing number of websites, like Amazon and the Internet Movie Database (IMBD), a website for movie reviews, allow users to publish reviews for things they are interested in, along with the growth of Web 2.0, where user interaction is prioritized. [13]

Online movie reviews are evolving into an important information source for users, with the continuous increase in data on the web [19]. However, online users post a significant number of movies reviews every day, hence making it difficult for them to manually summarize the reviews and determine their interest in the film. One of the challenging problems in natural language processing is mining and summarizing movie reviews. [13].

Authors' addresses: Nazhim Kalam, nazhimkalam@gmail.com, University of Westminster, 309 Regent St., London, UK; Torin Wirasingha, Informatics Institute of Technology, 57 Ramakrishna Rd, Colombo 06, Sri Lanka, torin.w@iit.ac.lk.

Text summary assist users or business decision-makers by compiling and analyzing a significant number of online reviews. [3]

These days, the majority of people research a film's reviews before selecting or watching it on any platform, such Netflix or Amazon Prime, but we also come across conflicting reviews that can be either good or bad. While most reviews are detailed and require a significant amount of time to review, this develops a problem where users aren't able to make quicker decisions. Therefore, by summarizing the review makes it easier and faster for users to make decisions. This can also help streaming services like Netflix quickly discover the viewing habits or preferences of their users [7]

2 BUSINESS CORPORATE ADVANTAGE

It is also known that it costs at least five times as much time and money to acquire a new customer as it does to keep an existing one, so it is important to learn how to foster customer loyalty to the brand, business, or service that is being offered. Customer satisfaction is essential to the survival of corporate industries. Understanding client expectations through their feedback or reviews helps business industries grow and fix faults [22]

On the other hand, companies like Netflix or Amazon Prime can use movie summaries to help users understand their watching pattern or their interest. Likewise, movie-related industries need to allow customers to quickly scan the summary and quickly decide whether they should be watching it or not [13]

3 TEXT SUMMARIZATION AND TECHNIQUES

With the massive accumulation of information/data on the internet nowadays, it is extremely difficult to extract relevant information from a large number of textual documents. The goal of text summarizing is to provide a condensed yet meaningful version of lengthy textual content [23] We all know that text summarization has several uses in a variety of internet-based fields, including search engines that are used for querying and e-commerce sites that utilize sentiment analysis to determine client satisfaction with items [10]

However, in the movie industry, consumers may utilize text summarization to simplify customer reviews of movies, which are often lengthy and time-consuming to read. This enables users to make better decisions when they decide whether or not to watch a certain movie [13]

Generally, text summarization is classified into two which are; abstractive text summarization and extractive text summarization, however, the approach for creating a hybrid model for text summarization is possible [3]. The abstractive text summarization technique aims to produce sentences on its own and then uses them to provide a coherent summary. Therefore, the summary's content will vary from the original context yet still convey the same idea [8]. Additionally, it is well-recognized that a strong abstractive summary encompasses the input's key details and is linguistically fluent [27].

The extractive text summarizing method focuses on picking out key phrases or groups of phrases from the original input content and combining them to produce a concise yet insightful text summary. It is determined which sentences should be included as parts of the summary based on the statistical and linguistic characteristics of the sentences [11] A hybrid system is one that combines various strategies to produce a single system. However, hybrid text summarizing systems do exist, for instance, using a combination of extractive and abstractive summarization can be utilized to generate a hybrid system that uses encoder-decoders [14]

4 NLP WITH DEEP LEARNING

NLP is a method for computers to intelligently and effectively analyze, comprehend, and derive meaning from human language, as opposed to other approaches that only focus on the interactions between human language and computers. Deep learning techniques are increasingly being used in the field of AI compared to traditional machine learning approaches due to their success rates in handling difficult high-computing learning tasks [18]

In today's NLP, machine learning is prominent, but for the most part, it only involves numerically optimizing the weights of characteristics and representations that have been created by humans. Deep learning aims to investigate how computers can utilize data to create features and representations suitable for challenging interpretation tasks [24].

5 NLP TRANSFORMER MODELS

Open-source library Transformers contains modern transformer architectures that have been thoroughly developed and are integrated by a common API. Pretraining has enabled the efficient use of this capacity for a wide range of activities, and these designs have permitted the construction of higher-capacity models. Transformers are designed to be easy for practitioners, expandable for researchers, and quick and reliable in industrial deployments [26]

It has been demonstrated that the modern generation of pre-trained language models based on transformers is rather competent at identifying syntactic signals like noun modifiers, possessive pronouns, prepositions, or co-referents, as well as semantic cues like entities and relations [6]

Hugging Face Hub offers a variety of transformer designs, including BERT, GPT2, T5, PEGASUS, and many others. The figure below represents the daily average for unique downloads of the pretrained transformer model architectures between Oct 2019 to May 2020 [26]

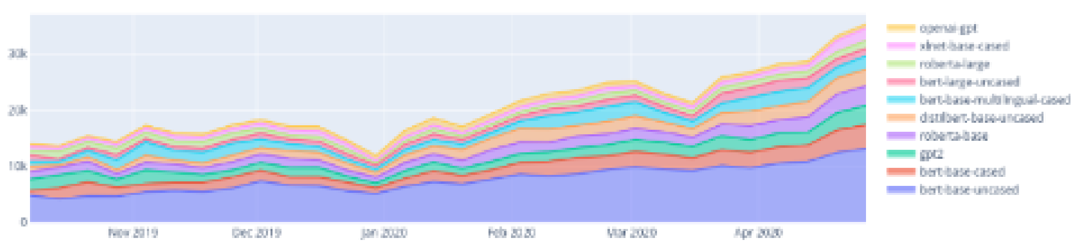


Fig. 1. Transformer Architecture Downloads Rate [26]

[10] research compares various other researchers approaches taken in order to perform abstractive text summarization, these techniques includes the use of transformers and other neural network approaches such as CNN and LSTM RNN networks. The research comparison table below only includes the approaches of transformers used taken abstractive text summarization.

6 AUTOMATED HYPERPARAMETER MODEL TUNING

Finding the ideal collection of parameter values to train an algorithm using in order to build a model relevant to the dataset is known as hyperparameter tuning [17]. The calculation of the performance improvement that may be obtained by changing the value of each of the considered hyperparameters from the original value to the value indicated in the target configuration set by the tuning strategy is where hyperparameters make the biggest contribution to improving algorithm performance [12]

There are several hyperparameters that play a significant role in performance enhancement; however, not all of the parameters do so; just a select handful do, for example, learning rate, weight decay, number of epochs, batch size, and warm up ratio. As a result, giving critical hyperparameters a higher priority is crucial [9]

Automated framework tools, such as Optuna, an open-source framework for hyperparameter optimization built on the Python programming language, does hyperparameter tweaking. The application of numerous hyperparameter optimization techniques, including Grid Search, Random Search, TPE, and CMA-ES algorithms, was made easier by this framework [12]

7 MODEL GENERALIZATION

Generalization now plays a significant part in resolving issues in numerous fields that are linked to the same issue. The capacity of a model to generalize to new, previously unobserved data that comes from the same distribution as the model's original data is known as generalization [2]

Generalization is a useful strategy for starting with the foundation and improving or specializing in one's field as more unseen domain data becomes available. Therefore, the generalized solution will be able to adapt to even unseen domain data, making this solution to solve a common problem in multiple domain [1]

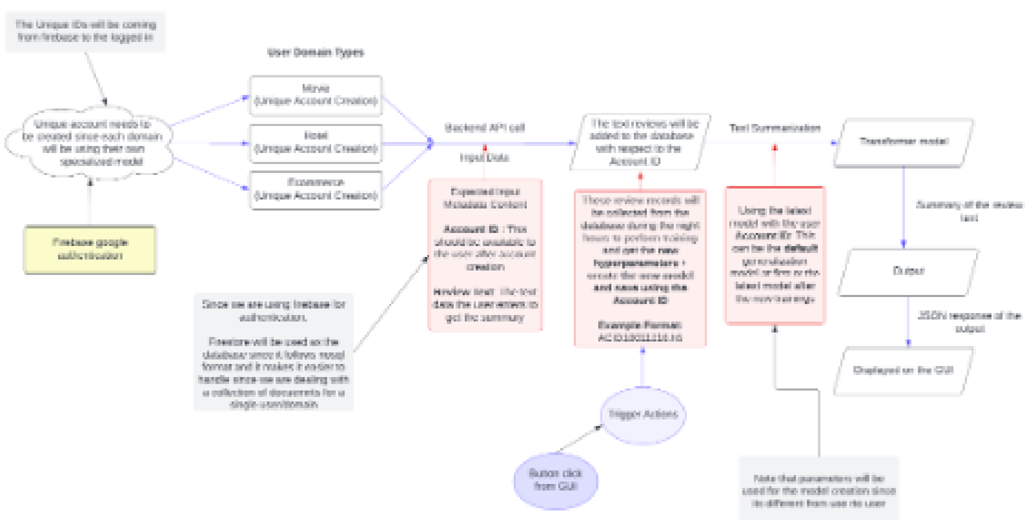


Fig. 2. Proposed Generalized Abstractive Summarization System Process Flow *(self-composed)*

8 ALGORITHMIC APPROACHES FOR TEXT SUMMARIZATION

The study of [13] starts by first focusing on feature extraction, then transforming reviews into vector spaces, and applying the Naive Bayes machine learning method for review classification utilizing an undirected weighted graph-based ranking algorithm to rank score for each review phrase in graph and then, in order to construct the extractive summary, the highest scoring sentences are selected. However, the author has limited the use of sophisticated deep learning algorithms to improve performance by solely using standard machine learning approaches to tackle the problem.

[6] research made use of seq2seq model for text summarization along with the attention mechanism for improved accuracy and the Concept net Number batch word embedding model, which is superior than Glove. Utilizing a 1D convolutional layer, a max pooling layer, an LSTM layer, and finally a fully connected layer at the very end. However, the author's use of generic deep learning algorithms to handle this problem introduces a new constraint that prevents performance from being improved using the most recent deep learning strategy for NLP-related problems, transformers.

The research of [21] liked mentioned earlier is an extractive method text summarization based on integer linear programming (ILP [Unsupervised method]) to choose an informative subset of opinions centered on the identified aspects. Utilize ROUGE-based criteria to assess and contrast the summaries and get competitive outcomes. Since the dataset is also constrained, extractive summaries could not be particularly insightful; thus, utilizing an abstractive technique might produce superior results, despite the dataset's constrained size.

The study of [10] focus of the authors' study is utilizing the encoder-decoder model with the attention layer to produce text summaries with good syntax and no repeated words. the creation of an encoder-decoder model with gated recurrent units and training it to provide an abstract summary of a piece of writing. Although the author employed deep learning, its application in production

9 USAGE OF TRANSFORMERS

[11] research employed pretrained models such Pipeline BART, BART modified, T5, and PEGASUS to deal with text summarization as a part of the comparison study done. The ROUGE Scores were used as the evaluation measures. During the experiments, the author employed transformer designs; however, the hyperparameters used were default and might be tuned for a better performance. The constraints consist of concentrating on developing more reliable models that can further expand the method to produce summaries of varying length and applicable for multi-document summarization.

[10] The author explores with deep learning methods in the broad text summarization domain to determine which method—among a collection that includes RNN, CNN, and Transformers—performs best. The author also considers metrics for model evaluations including BLEU and ROUGE, despite using sophisticated deep learning algorithms, the author was unable to undertake hyperparameter tuning to improve the method and obtain a better outcome.

10 MACHINE LEARNING TEXT SUMMARIZATION TECHNIQUES

[5] points out a previous research where a system was built that uses a hybrid classifier approach with machine learning algorithm combination of SVM and Naïve Bayes in sync with fuzzy logic and they also concluded that with the increase in the classifier count the accuracy can also be increased. They also made use of supervised ML algorithms such as KNN for the classification of the reviews which then combining appropriate words for identifying the features of the product.

[13] proposed system was for the movies domain using the customer reviews, the author broke down proposed methodology into segments of which is preprocessing, feature extraction, review classification and finally review summarization. The Nave Bayes (NB) classification method, which is regarded as a robust classifier and may achieve greater accuracy, was used to categorize the reviews from negative to positive using supervised ML classification technique, It is clear that an extractive summarization approach was used because the text summarization phase was completed in several stages, starting with the creation of a graph from classified reviews, followed by the ranking of graph nodes and the selection of the top rank sentences for the summary generation.

Initially, these machine learning methodologies were given a lot of significance, but as time has progressed on, new technologies and techniques have emerged that can utilize deep learning techniques like RNN, CNN, etc. to perform better.

11 DEEP LEARNING TEXT SUMMARIZATION TECHNIQUES

Numerous studies have been conducted on deep learning methods for abstractive text summarization, such as with the usage of CNN, LSTM-CNN, Convolutional Seq2Seq, Sequence to Sequence RNN, Convolutional Sequence to Sequence, Transformers, T5, BART, BERT etc.… which were trained on a general dataset such as from Gigaword, DUC 2002, DUC 2004, CNN Daily Mail, DUC, Xsum, Newsroom such datasets, in order to get an evaluation comparison on which outperforms the rest and eventually the T5 Transformer outperformed the rest of the other techniques in the case of abstractive text summarization [10]

[23] has conducted a thorough analysis of latest developments in seq2seq models for the task of abstractive text summarizing. The author's analysis includes a full review of several distinct seq2seq models for abstractive summarization.

Out of which transformers are the advanced deep learning approach for text summarization which is an encoder-decoder model with attention layer which helps it to generate better results than a traditional simple RNN architecture [10]

12 AVAILABLE DATASETS FOR GENERALIZED TEXT SUMMARIZATION

There are two datasets that the author will be exploring throughout the development of this project. One of which is the Amazon movie reviews dataset from Stanford University Education, which contains data within the span period of more than 10 years including 8 million review data records [20]

This dataset will be used to test out the solution for the problem domain which is abstractive text summarization for movies. Given that the author is able to create the solution for the domain of movies then, the author then plans to generalize the solution using another dataset named as Gigaword which is from TensorFlow datasets which was used previously for creating generalized content for text summarization [15]

13 PREPROCESSING TECHNIQUES USED IN TEXT SUMMARIZATION.

Text preprocessing is very important when it comes to dealing with text related data. In earlier studies, a variety of text preprocessing approaches were utilized for text summarization.

Sentence segmentation is a fundamental step in NLP applications including IR, machine translation, semantic role labeling, and summarization. It is the process of identifying boundaries within a document that divides the document's text into sentences, typically from a strong point of punctuation like (full stop, explanation mark, question mark, etc.), Tokenization and stop words removal will then be performed. Tokenization will be carried out by the tokenizer program to split the sentences into distinct words by splitting them at whitespaces such as blanks, tabs, and any strong punctuation. Stop word removal is also used to remove frequently used words in the document such as "I," "an," and "a" because these words carry little meaning and are best removed from the document [13]

Other researchers have incorporated a variety of other techniques, including noise removal, which eliminates unnecessary text from the input document, such as the header and footer, and named entity recognition (NER), which recognizes words in the input text as names of things like people, places, and things, among others [4]

Datasets may also contain unwanted records, null records, or redundant records that are absolutely useless. These records or rows with null values are eliminated, unnecessary HTML tags and URL links are also filtered off from the text as a part of text preprocessing. Contraction mapping is crucial and this will be handling which are converting short word formats into longer such as "aren't" into "are not". Converting the entire text content into a single case most preferably to lowercase, therefore further character filtration would become very simpler [10]

14 EVALUATION TECHNIQUES

A machine learning model's performance, as well as its advantages and disadvantages, are understood through the process of model evaluation, which employs many evaluation measures. During the early stages of research, it's critical to evaluate models to determine their efficiency. The table below shows the available measure and the metrics that can been used to quantitatively evaluate the text summarization system.

Table 1. Evaluation techniques for abstractive text summarization

Measure	Description	Objective Orientation
ROUGE	Measures are made by comparison between an automatically generated summary/translation against a group of reference summaries (generally human created summaries)	Positively oriented. Higher, the better.
BLEU	measures the precision (as to how much words in the generated summaries appeared in the human generated summaries)	

ROUGE also known as Recall-Oriented Understudy for Gisting Evaluation. Measures are made by comparison between an automatically generated summary/translation against a group of reference summaries (generally human created summaries) [16]. ROUGE measures the recall, (according to how frequently the terms from the summaries created by humans appeared in those computers - generated.)

BLEU also known as Bilingual Evaluation Understudy is a metric used for evaluation for the quality of machine generated text by comparing it with a reference text that is supposed to be generated. [25]. BLEU measures the precision (as to how much words in the generated summaries appeared in the human generated summaries)

15 CONCLUSION

In this review, the author has pointed out the reasons for the need for the need of performance increase for abstractive review summarization and how NLP transformers can be used as the solution.

Moreover the author also discusses how and why the solution is generalized along with the ways in making using automated hyperparameter tools and model retraining, evaluations metrics is also included by the author along with their objected orientations.

REFERENCES

[1] [n. d.] [pdf] domain generalization with mixstyle. (). https://www.semanticscholar.org/reader/4f6eafafc9563a5b904535078df7c74afe39ef59.

[2] [n. d.] [pdf] exploring generalization in deep learning. (). https://www.semanticscholar.org/reader/d53fb3feecab07a0d70bf466dd473ec6052ecc07.

[3] Alaa F. Alsaqer and Sreela Sasi. 2017. Movie review summarization and sentiment analysis using rapidminer. *2017 International Conference on Networks amp; Advances in Computational Technologies (NetACT)*. DOI: 10.1109/netact.2017.8076790.

[4] Nasid Habib Barna and Hasnain Heickal. 2022. An automatic abstractive text summarization system. *Dhaka University Journal of Applied Science and Engineering*, 6, 2, 39–48. DOI: 10.3329/dujase.v6i2.59217.

[5] Ravali Boorugu and G. Ramesh. 2020. A survey on nlp based text summarization for summarizing product reviews. *2020 Second International Conference on Inventive Research in Computing Applications (ICIRCA)*. DOI: 10.1109/icirca48905.2020.9183355.

[6] Adrian M. Brasoveanu and Razvan Andonie. 2020. Visualizing transformers for nlp: a brief survey. *2020 24th International Conference Information Visualisation (IV)*. DOI: 10.1109/iv51561.2020.00051.

[7] Kia Dashtipour, Mandar Gogate, Ahsan Adeel, Hadi Larijani, and Amir Hussain. 2021. Sentiment analysis of persian movie reviews using deep learning. *Entropy*, 23, 5, 596. DOI: 10.3390/e23050596.

[8] Shivam Duseja. 2020. Text summarization using deep neural networks. (Aug. 2020). https://towardsdatascience.com/text-summarization-using-deep-neural-networks-e7ee7521d804.

[9] Sylvia Engdahl. 2008. Blogs. (2008). https://aws.amazon.com/blogs/machine-learning/bring-your-own-hyperparameter-optimization-algorithm-on-amazon-sagemaker/.

[10] Abdul Ghafoor Etemad, Ali Imam Abidi, and Megha Chhabra. 2021. A review on abstractive text summarization using deep learning. *2021 9th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO)*. DOI: 10.1109/icrito51393.2021.9596500.

[11] Vishal Gupta and Gurpreet Singh Lehal. 2010. A survey of text summarization extractive techniques. *Journal of Emerging Technologies in Web Intelligence*, 2, 3. DOI: 10.4304/jetwi.2.3.258-268.

[12] Johnsymol Joy and Mercy Paul Selvan. 2022. A comprehensive study on the performance of different multi-class classification algorithms and hyperparameter tuning techniques using optuna. *2022 International Conference on Computing, Communication, Security and Intelligent Systems (IC3SIS)*. DOI: 10.1109/ic3sis54991.2022.9885695.

[13] Atif Khan, Muhammad Adnan Gul, Mahdi Zareei, R. R. Biswal, Asim Zeb, Muhammad Naeem, Yousaf Saeed, and Naomie Salim. 2020. Movie review summarization using supervised learning and graph-based ranking algorithm. *Computational Intelligence and Neuroscience*, 2020, 1–14. DOI: 10.1155/2020/7526580.

[14] Mahira Kirmani, Nida Manzoor Hakak, Mudasir Mohd, and Mohsin Mohd. 1970. Hybrid text summarization: a survey. (Jan. 1970). http://link.springer.com/10.1007/978-981-13-0589-4_7.

[15] Panagiotis Kouris, Georgios Alexandridis, and Andreas Stafylopatis. 2019. Abstractive text summarization based on deep learning and semantic content generalization. *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. DOI: 10.18653/v1/p19-1501.

[16] Chin-Yew Lin. [n. d.] Rouge: a package for automatic evaluation of summaries. (). https://aclanthology.org/W04-1013.

[17] Xueqing Liu and Chi Wang. 2021. An empirical study on hyperparameter optimization for fine-tuning pre-trained language models. (June 2021). http://arxiv.org/abs/2106.09204.

[18] Marc Moreno Lopez and Jugal Kalita. 2017. Deep learning applied to nlp. (Mar. 2017). http://arxiv.org/abs/1703.03091.

[19] Mamtesh M and Seema Mehla. 2019. Sentiment analysis of movie reviews using machine learning classifiers. *International Journal of Computer Applications*, 182, 50, 25–28. DOI: 10.5120/ijca2019918756.

[20] Julian John McAuley and Jure Leskovec. 2013. From amateurs to connoisseurs. *Proceedings of the 22nd international conference on World Wide Web*. DOI: 10.1145/2488388.2488466.

[21] Rajdeep Mukherjee, Hari Chandana Peruri, Uppada Vishnu, Pawan Goyal, Sourangshu Bhattacharya, and Niloy Ganguly. 2020. Read what you need: controllable aspect-based opinion summarization of tourist reviews. *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. DOI: 10.1145/3397271.3401269.

[22] Abraham Pizam and Taylor Ellis. 1999. Customer satisfaction and its measurement in hospitality enterprises. *International Journal of Contemporary Hospitality Management*, 11, 7, 326–339. DOI: 10.1108/09596119910293231.

[23] Tian Shi, Yaser Keneshloo, Naren Ramakrishnan, and Chandan K. Reddy. 2020. Neural abstractive text summarization with sequence-to-sequence models. (Sept. 2020). https://arxiv.org/abs/1812.02303.

[24] Richard Socher, Yoshua Bengio, and Christopher D. Manning. [n. d.] Deep learning for nlp (without magic). (). https://aclanthology.org/P12-4005.

[25] Josef Steinberger and Karel Ježek. [n. d.] Evaluation measures for text summarization. (). http://www.cai.sk/ojs/index.php/cai/article/view/37.

[26] Thomas Wolf 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*. DOI: 10.18653/v1/2020.emnlp-demos.6.

[27] Haoyu Zhang, Jianjun Xu, and Ji Wang. 2019. Pretraining-based natural language generation for text summarization. (Apr. 2019). https://arxiv.org/abs/1902.09243.

APPENDIX K – RESEARCH PAPER

An Adaptive Approach for Generalized Text Summarization using Optimized Transformers

Nazhim Kalam
Computer Science and Engineering
University of Westminster
London, UK
nazhimkalam@gmail.com

Torin Wirasingha
Department of Computing
Informatics Institute of Technology
Colombo, Sri Lanka
torin.w@iit.ac.lk

***Abstract*—This research explores the ways in which to get an optimal solution using Transformer for abstractive text summarization and yet making a generalized solution that can be adapted with respect to any domain (be it hotels, movies, restaurants) and increase its performance as the system gets used over time.**

***Index Terms*—Natural Language Processing, Machine Learning, Deep Learning, Recall-Oriented Understudy for Gisting Evaluation, Inductive logic programming**

I. Introduction

Today, there is a lot of textual material available, including news stories and reviews. Text summarizing helps us quickly find the key elements of the full piece by minimizing the quantity of text. [1]. Transformers in NLP is a novel architecture that aims to solve sequence-to-sequence tasks while handling long-range dependencies with ease. It has surpassed competing neural models like CNN (Convolutional Neural Nets) and RNN (Recurrent Neural Nets) in terms of performance to appear as the dominant architecture for natural language processing [2].

A. User Reviews

A user/customer review is typically referred to be written feedback from a customer who has used a product or service. Consumers frequently use user ratings and reviews to drive their purchasing decisions. Because the review data is unstructured, it becomes more challenging for consumers to compare and understand lengthier reviews [3]

User and customer reviews are extremely important to major corporations like tourism and hospitality, as they constitute the primary engine for the country's economic growth and development. where tourists from over the world may blog about their experiences and share their reviews online in numerous formats [4].

B. Corporate Advantage

It is also known that it costs at least five times as much time and money to acquire a new customer as it does to keep an existing one, so it is important to learn how to foster customer loyalty to the brand, business, or service that is being offered. Customer satisfaction is essential to the survival of corporate industries. Understanding client expectations through their feedback or reviews helps business industries grow and fix faults [5].

On the other hand, companies like Netflix or Amazon Prime can use movie summaries to help users understand their watching pattern or their interest. Likewise, movie-related industries need to allow customers to quickly scan the summary and quickly decide whether they should be watching it or not [6].

C. Text Summarization

With the massive accumulation of information/data on the internet nowadays, it is extremely difficult to extract relevant information from numerous textual documents. The goal of text summarizing is to provide a condensed yet meaningful version of lengthy textual content [7].

We all know that text summarization has several uses in a variety of internet-based fields, including search engines that are used for querying and e-commerce sites that utilize sentiment analysis to determine client satisfaction with items [1].

However, in the movie industry, consumers may utilize text summarization to simplify customer reviews of movies, which are often lengthy and time-consuming to read. This enables users to make better decisions when they decide whether to watch a certain movie [6].

D. Abstractive and Extractive Techniques

Generally, text summarization is classified into two which are; abstractive text summarization and extractive text summarization, however, the approach for creating a hybrid model for text summarization is possible [8]. The abstractive text summarization technique aims to produce sentences on its own and then uses them to provide a coherent summary. Therefore, the summary's content will vary from the original context yet still convey the same idea [1]. Additionally, it is well-recognized that a strong abstractive summary encompasses the input's key details and is linguistically fluent [9].

The extractive text summarizing method focuses on picking out key phrases or groups of phrases from the original input content and combining them to produce a concise yet insightful text summary. It is determined which sentences should be included as parts of the summary based on the statistical and linguistic characteristics of the sentences [10]. A hybrid system is one that combines various strategies to produce a single system. However, hybrid text summarizing systems do exist, for instance, using a combination of extractive and abstractive summarization can be utilized to generate a hybrid system that uses encoder-decoders [11]

E. NLP with Deep Learning

NLP is a method for computers to intelligently and effectively analyze, comprehend, and derive meaning from human language, as opposed to other approaches that only focus on the interactions between human language and computers. Deep learning techniques are increasingly being used in the field of AI compared to traditional machine learning approaches due to their success rates in handling difficult high-computing learning tasks [12].

In today's NLP, machine learning is prominent, but for the most part, it only involves numerically optimizing the weights of characteristics and representations that have been created by humans. Deep learning aims to investigate how computers can utilize data to create features and representations suitable for challenging interpretation tasks [13].

F. Transformers

The open-source library Transformers contain modern transformer architectures that have been thoroughly developed and are integrated by a common API. Pretraining has enabled the efficient use of this capacity for a wide range of activities, and these designs have permitted the construction of higher-capacity models. Transformers are designed to be easy for practitioners, expandable for researchers, and quick and reliable in industrial deployments [2].

It has been demonstrated that the modern generation of pre-trained language models based on transformers is rather competent at identifying syntactic signals like noun modifiers, possessive pronouns, prepositions, or co-referents, as well as semantic cues like entities and relations [14]. Hugging Face Hub offers a variety of transformer designs, including BERT, GPT2, T5, PEGASUS, and many others. The figure below represents the daily average for unique downloads of the pre-trained transformer model architectures between Oct 2019 to May 2020 [2].

[1] the research compares various other researchers' approaches taken in order to perform abstractive text summarization, these techniques include the use of transformers and other neural networks approach such as CNN and LSTM RNN networks. The research comparison table below only includes the approaches of transformers used, taken in abstractive text summarization.

G. Hyperparameter Tuning

Finding the ideal collection of parameter values to train an algorithm using in order to build a model relevant to the dataset is known as hyperparameter tuning [15]. The calculation of the performance improvement that may be obtained by changing the value of each of the considered hyperparameters from the original value to the value indicated in the target configuration set by the tuning strategy is where hyperparameters make the biggest contribution to improving algorithm performance [16].

There are several hyperparameters that play a significant role in performance enhancement; however, not all the parameters do so; just a select handful do, for example, learning rate, weight decay, number of epochs, batch size, and warm-up ratio. As a result, giving critical hyperparameters a higher priority is crucial [17].

Automated framework tools, such as Optuna, an open-source framework for hyperparameter optimization built on the Python programming language, do hyperparameter tweaking. The application of numerous hyperparameter optimization techniques, including Grid Search, Random Search, TPE, and CMA-ES algorithms, was made easier by this framework [16].

H. Generalization

Generalization now plays a significant part in resolving issues in numerous fields that are linked to the same issue. The capacity of a model to generalize to new, previously unobserved data that comes from the same distribution as the model's original data is known as a generalization [1].

Generalization is a useful strategy for starting with the foundation and improving or specializing in one's field as more unseen domain data becomes available. Therefore, the generalized solution will be able to adapt to even unseen domain data, making this solution solve a common problem in multiple domains [18].

I. Data Expansion

The quality of a machine learning or deep learning model depends on a number of factors, one of which is the amount and quality of data fed during model training. There are several approaches to increase or expand your available data, one of which is data augmentation (making use of existing data points to create new data points). Making use of new data from the user end by saving as the model is used is another way of exposing new data for model retraining [19]. When generalized models are required to adapt to become domain-specific, model retraining will be considered with new data used by the specialized domain as the application is used.

II. Motivation to Enhance Abstractive Text Summarization Performance

The identified problem can also be applied to several other domains which require improving the quality of abstractive text summarization using the advanced approaches of deep learning, not only specific movie reviews, this is why a generalized solution was thought of initially [20].

As mentioned in the work of [1], syntactic and semantic issues with text summarization were the main issues that researchers were concerned with solving, and with respect to their research by exploring multiple deep learning techniques, they concluded that Transformer based models (T5 model) outperformed in all NLP tasks, this encourages the author to go deeper into the field of transformers optimization in order to enhance the quality of text summarization and address the constraints associated with the summarizing of movie reviews.

In the domain of movie review summarization, currently, there is no research done using the latest deep learning approaches (such as Transformers) to solve this problem, standard machine & deep learning algorithms such as Naïve Bayes, and RNN has been used, and the usage of advanced deep learning approaches can be utilized in order to enhance the quality/accuracy of the text summarization.

Deep learning models take longer to train, but they provide greater accuracy since they can simultaneously automate feature extraction and classification, whereas machine learning algorithms require feature selection at first. Therefore, applying deep learning techniques will help to improve the quality of text summarization and help the user in making better decisions [1].

III. Existing Work

A. *Text Summarization Systems*

There were multiple studies done previously in the area of text summarization, regarding both abstractive and extractive text summarization. [6] research is related to the domain of movie reviews summarization which is the same as this project domain, where the author has developed an automatic approach to summarize lengthy movie reviews along with a feature where the users are allowed to quickly recognize the positive and negative aspects of the movie with respect to the review process with. The text summarization approach taken by the author is an extractive approach, where sentence score ranking plays a major role in creating the summary.

The study of [21] is towards the domain of e-commerce but yet related to text summarization for customer reviews on the products they sell, so the purpose is that allow other customers to make better purchasing decisions on products, therefore the hassle of going through all the reviews to making a purchasing decision can be reduced to save time, the abstractive approach is considered to create the summary, which is a better choice of approach.

The research of [4] is another extractive approach for text summarization, where the author develops a solution for generating personalized aspect-based opinion summaries using a dataset that consists of a large collection of online tourist reviews. In addition, the author has gone a step further to personalize the summary's qualities by using the user's interest. However, using abstractive summarization would be a more effective strategy but also challenging when user interest customization is considered because the sentences have been created using their own words rather than with any sentence ranking technique.

[10] research is a comprehensive comparison study with benchmarking results of various pre-trained transformer architectures such as BART, BERT, T5, PEGASUS, etc... for abstractive text summarization which is an abstractive approach. This study includes the various types of datasets used to explore each model, with the evaluations as benchmarking results. The author has also concluded the best-performing transformer architecture as T5 by comparing the evaluation results of the study.

The study conducted by [1] is also an abstractive approach to text summarization with the addition of proper grammar and no repeated words used a deep learning approach with RNN and likewise [1] research also relates to an experimenting study with various deep learning approaches for abstractive text summarization along with the evaluation benchmarking with a goal in search for the best deep learning approach for the problem.

B. *Algorithmic approaches for Text Summarization*

The study of [6] starts by first focusing on feature extraction, then transforming reviews into vector spaces, and applying the Naive Bayes machine learning method for review classification utilizing an undirected weighted graph-based ranking algorithm to rank score for each review phrase in graph and then, in order to construct the extractive summary, the highest scoring sentences are selected. However, the author has limited the use of sophisticated deep learning algorithms to improve performance by solely using standard machine learning approaches to tackle the problem.

[21] research made use of the seq2seq model for text summarization along with the attention mechanism for improved accuracy and the concept net number batch word embedding model, which is superior to the glove. Utilizing a 1D convolutional layer, a max pooling layer, an LSTM layer, and finally a fully connected layer at the very end. However, the author's use of generic deep learning algorithms to handle this problem introduces a new constraint that prevents performance from being improved using the most recent deep learning strategy for NLP-related problems, transformers.

The study of [1] focuses on the author's study utilizing the encoder-decoder model with the attention layer to produce text summaries with good syntax and no repeated words. The creation of an encoder-decoder model with gated recurrent units and training it to provide an abstract summary of a piece of writing. Although the author employed deep learning, its application in production required real-time training so that it could be updated with the most recent content over time.

C. *Usage of Transformers*

[10] The research employed pre-trained models such as Pipeline BART, BART modified, T5, and PEGASUS to deal with text summarization as a part of the comparison study done. The ROUGE Scores were used as the evaluation measures. During the experiments, the author employed transformer designs; however, the hyperparameters used were defaulted and might be tuned for better performance. The constraints consist of concentrating on developing more reliable models that can further expand the method to produce summaries of varying lengths and applicable for multi-document summarization.

[1] The author explores deep learning methods in the broad text summarization domain to determine which method—among a collection that includes RNN, CNN, and

Transformers—performs best. The author also considers metrics for model evaluations including BLEU and ROUGE, despite using sophisticated deep learning algorithms, the author was unable to undertake hyperparameter tuning to improve the method and obtain a better outcome.

IV. Proposed Solution Approach

The diagram below shows how the approach for creating a generalized model is considered and executed using the optimized transformer models.

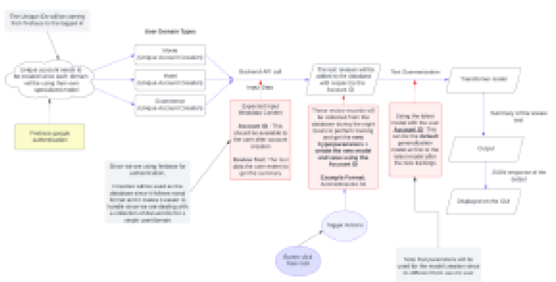


Fig. 1. Proposed Generalized Abstractive Summarization System Architecture

V. Evaluation

A machine learning model's performance, as well as its advantages and disadvantages, are understood through the process of model evaluation, which employs many evaluation measures. During the early stages of research, it's critical to evaluate models to determine their efficiency.

ROUGE also known as Recall-Oriented Understudy for Gisting Evaluation. Measures are made by comparison between an automatically generated summary/translation against a group of reference summaries (generally human-created summaries) [22]. ROUGE measures the recall, (according to how frequently the terms from the summaries created by humans appeared in those computers - generated.)

$$ROUGE-1 = \frac{(\sum_{Reference\ Summary} \sum_{unigram} Count_{match}(unigram))}{(\sum_{Reference\ Summary} \sum_{unigram} Count(unigram))}$$

Fig. 2. Rouge-1 Score Equation

$$ROUGE-2 = \frac{\sum_{s\in\{RefSummaries\}} \sum_{bigrams\ i\in S} \min(count(i,X), count(i,S))}{\sum_{s\in\{RefSummaries\}} \sum_{bigrams\ i\in S} count(i,S)}$$

Fig. 3. Rouge-2 Score Equation

$$ROUGE-L_{(candidate, references)} = max_k\{ROUGE-L_{single}(candidate, references_k)\}$$

Fig. 4. Rouge-L Score Equation

BLEU also known as Bilingual Evaluation Understudy is a metric used for the evaluation of the quality of machine-generated text by comparing it with a reference text that is supposed to be generated. [23]. BLEU measures the precision (as to how many words in the generated summaries appeared in the human-generated summaries)

$$BLEU = \frac{Number\ of\ words\ in\ the\ summary\ which\ are\ in\ gold\ standard}{Total\ number\ of\ words\ in\ the\ summary}$$

Fig. 5. BLEU Score Equation

The ROUGE score was used as the final evaluation metric for this research since the weightage of it is the best metric for this research as proven by previous work.

BART, T5 & Pegasus transformer architecture was experimented with for this research, out of which Bart gave the optimal results whereas T5 came close however Pegasus failed, there can be several reasons why Pegasus failed one of which can be due to the difference in the model architecture and the type of problem it can solve with respect to the dataset.

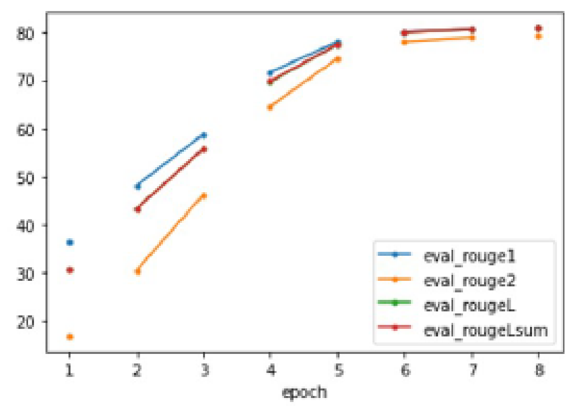


Fig. 6. Validation accuracy by the number of epochs - BART Model

ROUGE1 of 80.8, ROUGE2 of 79.42, ROUGE L of 80.8, and ROUGELSUM of 80.8 was the optimal evaluation metric result achieved from the BART model giving the best result.

VI. Future Enhancements

Future Enhancements for text summarization models can be achieved through the use of transformer hybridization. This approach could lead to improved performance, enabling the models to better handle complex natural language processing tasks. Another potential improvement is the inclusion of text paraphrase models for user reviews.

This would help to address the potential issue of inaccuracies in user-generated content. Additionally, applying keyword extraction for sentiment classification in review summaries could provide valuable insights for domain users to improve their services.

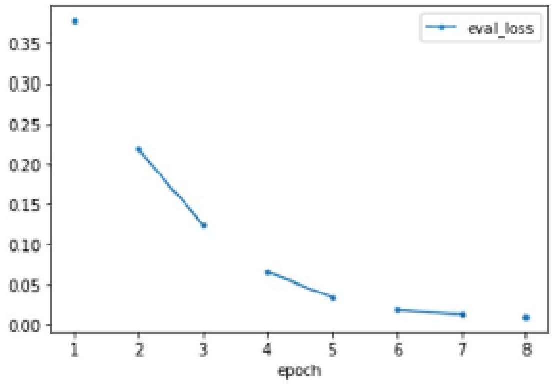


Fig. 7. Validation loss by the number of epochs - BART Model

By identifying which keywords contributed to the sentiment classification, businesses can understand their customers better and make informed decisions to enhance their services. These advancements can significantly benefit the field of natural language processing and improve the accuracy and efficiency of text summarization models.

VII. Conclusion

The conclusion of this study finds that the author was able to design, build, and evaluate an adaptive generalized abstractive text summarization system using optimized transformers and automated hyperparameter tuning and model retraining with respect to any domain.

References

[1] A. G. Etemad, A. I. Abidi, and M. Chhabra, "A review on abstractive text summarization using deep learning," *2021 9th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO)*, 2021. DOI: 10.1109/icrito51393.2021.9596500.

[2] T. Wolf, L. Debut, V. Sanh, *et al.*, "Transformers: State-of-the-art natural language processing," *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, 2020. DOI: 10.18653/v1/2020.emnlp-demos.6.

[3] J. J. McAuley and J. Leskovec, "From amateurs to connoisseurs," *Proceedings of the 22nd international conference on World Wide Web*, 2013. DOI: 10.1145/2488388.2488466.

[4] R. Mukherjee, H. C. Peruri, U. Vishnu, P. Goyal, S. Bhattacharya, and N. Ganguly, "Read what you need: Controllable aspect-based opinion summarization of tourist reviews," *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2020. DOI: 10.1145/3397271.3401269.

[5] A. Pizam and T. Ellis, "Customer satisfaction and its measurement in hospitality enterprises," *International Journal of Contemporary Hospitality Management*, vol. 11, no. 7, pp. 326–339, 1999. DOI: 10.1108/09596119910293231.

[6] A. Khan, M. A. Gul, M. Zareei, *et al.*, "Movie review summarization using supervised learning and graph-based ranking algorithm," *Computational Intelligence and Neuroscience*, vol. 2020, pp. 1–14, 2020. DOI: 10.1155/2020/7526580.

[7] T. Shi, Y. Keneshloo, N. Ramakrishnan, and C. K. Reddy, *Neural abstractive text summarization with sequence-to-sequence models*, Sep. 2020. [Online]. Available: https://arxiv.org/abs/1812.02303.

[8] A. F. Alsaqer and S. Sasi, "Movie review summarization and sentiment analysis using rapidminer," *2017 International Conference on Networks amp; Advances in Computational Technologies (NetACT)*, 2017. DOI: 10.1109/netact.2017.8076790.

[9] H. Zhang, J. Xu, and J. Wang, *Pretraining-based natural language generation for text summarization*, Apr. 2019. [Online]. Available: https://arxiv.org/abs/1902.09243.

[10] V. Gupta and G. S. Lehal, "A survey of text summarization extractive techniques," *Journal of Emerging Technologies in Web Intelligence*, vol. 2, no. 3, 2010. DOI: 10.4304/jetwi.2.3.258-268.

[11] M. Kirmani, N. Manzoor Hakak, M. Mohd, and M. Mohd, *Hybrid text summarization: A survey*, Jan. 1970. [Online]. Available: http://link.springer.com/10.1007/978-981-13-0589-4_7.

[12] M. M. Lopez and J. Kalita, *Deep learning applied to nlp*, Mar. 2017. [Online]. Available: http://arxiv.org/abs/1703.03091.

[13] R. Socher, Y. Bengio, and C. D. Manning, *Deep learning for nlp (without magic)*. [Online]. Available: https://aclanthology.org/P12-4005.

[14] A. M. Brasoveanu and R. Andonie, "Visualizing transformers for nlp: A brief survey," *2020 24th International Conference Information Visualisation (IV)*, 2020. DOI: 10.1109/iv51561.2020.00051.

[15] X. Liu and C. Wang, *An empirical study on hyperparameter optimization for fine-tuning pre-trained language models*, Jun. 2021. [Online]. Available: http://arxiv.org/abs/2106.09204.

[16] J. Joy and M. P. Selvan, "A comprehensive study on the performance of different multi-class classification algorithms and hyperparameter tuning techniques using optuna," *2022 International Conference on Computing, Communication, Security and Intelligent Systems (IC3SIS)*, 2022. DOI: 10.1109/ic3sis54991.2022.9885695.

[17] S. Engdahl, *Blogs*, 2008. [Online]. Available: https://aws.amazon.com/blogs/machine-learning/bring-your-own-hyperparameter-optimization-algorithm-on-amazon-sagemaker/.

[18] *[pdf] domain generalization with mixstyle*. [Online]. Available: https://www.semanticscholar.org/reader/4f6eafafc9563a5b904535078df7e74afe39ef59.

[19] N. H. Barna and H. Heickal, "An automatic abstractive text summarization system," *Dhaka University Journal of Applied Science and Engineering*, vol. 6, no. 2, pp. 39–48, 2022. DOI: 10.3329/dujase.v6i2.59217.

[20] P. Kouris, G. Alexandridis, and A. Stafylopatis, "Abstractive text summarization based on deep learning and semantic content generalization," *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 2019. DOI: 10.18653/v1/p19-1501.

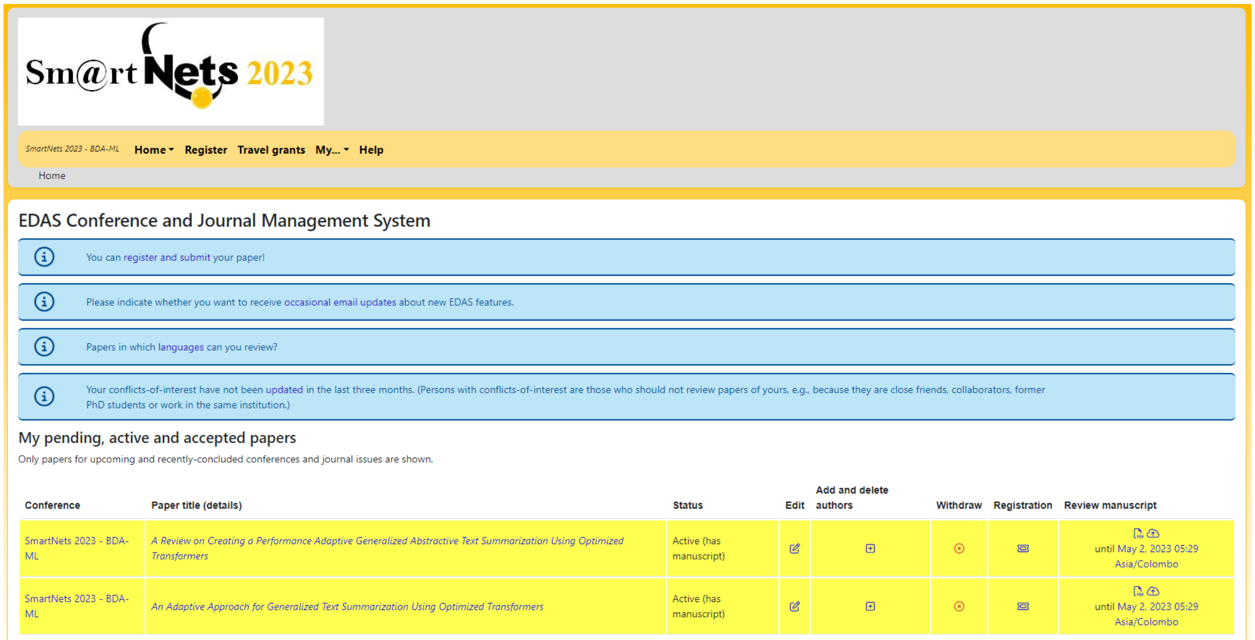
[21] R. Boorugu and G. Ramesh, "A survey on nlp based text summarization for summarizing product reviews," *2020 Second International Conference on Inventive Research in Computing Applications (ICIRCA)*, 2020. DOI: 10.1109/icirca48905.2020.9183355.

[22] C.-Y. Lin, *Rouge: A package for automatic evaluation of summaries*. [Online]. Available: https://aclanthology.org/W04-1013.

[23] J. Steinberger and K. Ježek, *Evaluation measures for text summarization*. [Online]. Available: http://www.cai.sk/ojs/index.php/cai/article/view/37.

APPENDIX L – PROOF OF SUBMISSION

Figure 61: Research & Review Paper Submission



The above figure shows the proof of the review and research paper submission at SmartNETs 2023 under EDAS Conference and Journal Management System.

END OF THESIS