NLP-Based AI Robot for Intelligent Interaction and Physical Library Support

by

Golam Dastagir
20101493
Khaled Mushahed Hossain
20101297
Md Mahfujul Haque
20101445
Raiyan Bin Gaffar
20101056

A thesis submitted to the Department of Computer Science and Engineering
in partial fulfillment of the requirements for the degree of
B.Sc. in Computer Science

Department of Computer Science and Engineering
Brac University
January 2024

© 2024. Brac University
All rights reserved.

Declaration

It is hereby declared that

1. The thesis submitted is my/our own original work while completing degree at Brac University.
2. The thesis does not contain material previously published or written by a third party, except where this is appropriately cited through full and accurate referencing.
3. The thesis does not contain material which has been accepted, or submitted, for any other degree or diploma at a university or other institution.
4. We have acknowledged all main sources of help.

Student's Full Name & Signature:

Golam Dastagir

20101493

Md Mahfujul Haque

20101445

Khaled Mushahed Hossain

20101297

Raiyan Bin Gaffar

20101056

Approval

The thesis/project titled "NLP-Based AI Robot for Intelligent Interaction and Physical Library Support" submitted by

1. Golam Dastagir (20101493)
2. Khaled Mushahed Hossain (20101297)
3. Md Mahfujul Haque (20101445)
4. Raiyan Bin Gaffar (20101056)

Of Fall, 2023 has been accepted as satisfactory in partial fulfillment of the requirement for the degree of B.Sc. in Computer Science on January 09, 2024.

Examining Committee:

Supervisor:
(Member)

Dr. Md. Khalilur Rahman
Professor
Department of Computer Science and Engineering
BRAC University

Program Coordinator:
(Member)

Dr. Md. Golam Rabiul Alam
Professor
Department of Computer Science and Engineering
BRAC University

Head of Department:
(Chair)

Dr. Sadia Hamid Kazi
Chairperson and Associate Professor
Department of Computer Science and Engineering
BRAC University

Ethics Statement

We pledge to uphold the legal requirements of our home nation, workplace, university, and community. We promise to behave honorably, faithfully, and with integrity. We will be willing to own up to our mistakes and only enter into agreements that we intend to keep.

Abstract

The benefits and advantages of integrating AI, ML, and NLP into library services which can redefine how traditional libraries function are thoroughly discussed in this paper. The practical applications of ML and AI in library settings are also explored here. With challenges like information overload, evolving user expectations, the need for personalized experiences, and other similar issues raise the proposal for the development of an advanced library assistant. In our study, we have gone through different academic publications as well as examining different scenarios in which these implications will be able to enhance library operations. Also, the introduction of a responsive bot system in our study further extends the Library Assistant's capabilities, which has made it possible to bridge the divide between physical and virtual library spaces. Our research comprehends the broad range of redefining the role of libraries in the digital age, with the envisioning towards a dynamic future where libraries will seamlessly blend tradition with innovation. This paper presents an in-depth analysis regarding how the system is able to enhance the library mechanism by providing perception into user sentiment as well as enabling an efficient book discovery. We have provided particular consideration to instructive case studies, project execution, and theoretical frameworks. The paper also emphasizes the unique role that is done by the librarians, acknowledging their ethical decision-making capabilities as well as emotional intelligence which is not possible by any robot or assistant system. By combining all of this intensive study, we hope to highlight the important influence that AI and NLP technologies may have on library operations. The proper combination of AI technologies could indeed contribute to major technical advances for libraries, putting them at the forefront of society's growth throughout this digital transition.

Keywords: Artificial Intelligence (AI), Machine Learning (ML), Natural Language Processing (NLP), Theoretical Frameworks, Case Studies

Dedication

Our thesis is dedicated to our family and friends, who have supported us throughout the whole process and provided us with inspiration to successfully finish our paper in difficult circumstances.

Acknowledgement

Firstly, all praise to the Great Allah for whom our thesis have been completed without any major interruption.

Secondly, to our supervisor Dr. Md. Khalilur Rahman sir for his kind support and advice in our work. He helped us whenever we needed help.

And finally to our parents without their throughout sup-port it may not be possible. With their kind support and prayer we are now on the verge of our graduation.

Table of Contents

Declaration **i**

Approval **ii**

Ethics Statement **iii**

Abstract **iv**

Dedication **v**

Acknowledgment **vi**

Table of Contents **vii**

List of Figures **ix**

List of Tables **x**

Nomenclature **xi**

1 Introduction **1**
- 1.1 Background and Motivation 1
- 1.2 Research Problem 2
- 1.3 Research Objective 3

2 Related Work **5**
- 2.1 Analysis of Qualities Needed for AI Machines in Libraries 5
- 2.2 Traditional Library Management Systems 5
- 2.3 Analysis of Automation in Libraries with AI 6
- 2.4 Integration of AI for Efficient Library Management 6
- 2.5 Potential Evolution of Smart Libraries 7
- 2.6 Emulation of Expert System for Library Support 7
- 2.7 Essential Techniques for Book Recommendations 7

3 Dataset and Preprocessing **9**
- 3.1 Dataset Collection and Creation 9
  - 3.1.1 Query to function Dataset 9
  - 3.1.2 Library Question and Answer 10
  - 3.1.3 Books 11
  - 3.1.4 Collaborative Book Recommendation System 12

3.1.5 Emotion Dataset . . . 12
3.2 Preprocessing . . . 13
3.2.1 Query to function Dataset . . . 13
3.2.2 Books . . . 13
3.2.3 Collaborative Book Recommendation System . . . 14
3.2.4 Emotion Dataset . . . 14

4 Proposed Methodology 17
4.1 Background . . . 17
4.1.1 Transformer based Models . . . 17
4.1.2 k-Nearest Neighbor(kNN) . . . 20
4.1.3 Singular Value Decomposition (SVD) . . . 22
4.1.4 Multinomial Naive Bayes (MultinomialNB) . . . 23
4.1.5 Term Frequency - Inverse Document Frequency (TF-IDF) . . . 23
4.2 Robot Model . . . 24
4.2.1 Robot system overview . . . 24
4.2.2 Robot Architechture . . . 25
4.3 Workplan . . . 27

5 Results, Analysis and Discussion 30
5.1 Chatbot accuracy . . . 30
5.1.1 Comparision between BERT, RoBERTa, and DistilBERT . . . 30
5.2 Summarizier accuracy . . . 32
5.2.1 Evaluation Model . . . 32
5.2.2 Comparision between BART, PEGASUS, T5 . . . 33
5.3 Recommendation accuracy . . . 35
5.4 Sentiment accuracy . . . 37
5.4.1 Analysis of Computer Vision System For Robot . . . 38

6 Conclusion and Future Work 41
6.1 Conclusion . . . 41
6.2 Future Work . . . 41

Bibliography 44

List of Figures

3.1 Variation of functions in the Chatbot Dataset 10
3.2 Variation of functions in Chatbot Dataset after preprocessing 13
3.3 Emotion Dataset Chart 15
3.4 Preprocessed Emotion Dataset Chart 16

4.1 MLM (left) + GSG (right) training together in PEGASUS 19
4.2 T5 Architecture 20
4.3 k Nearest Neighbor Algorithm 21
4.4 Singular Value Decomposition Algorithm 22
4.5 Robot Architecture 25
4.6 Necessary Equipments 26
4.7 Model Workflow 27
4.8 Chatbot Workflow 28
4.9 Work plan of the Robot 29

5.1 Example of the prediction capability of user queries 31
5.2 Comparision of the chatbot models 32
5.3 Comparision of BLEU Scores 33
5.4 Comparision of ROGUE Scores 34
5.5 Example of Summarizer function 34
5.6 RMSE comparison between SVD and kNN 36
5.7 MAE comparison between SVD and kNN 36
5.8 Performance Metrics of Sentiment Analysis 37
5.9 Predicting User Emotion 38
5.10 Detecting QR code coordinate at (99,100) - Signaling to go left. . . . 39
5.11 Detecting QR code coordinate at (320,74) - Signaling to go forward. . 40
5.12 Detecting QR code coordinate at (337,160) - Signaling steady. 40

List of Tables

3.1 Example of the Dataset for Chatbot 9
3.2 Category - Cost of Books 10
3.3 Category - Library Section 11
3.4 Category - Library information 11
3.5 Ebook available for Summary and TTS 11
3.6 Book Information 12
3.7 Ratings 12
3.8 Emotion Dataset 12
3.9 Preprocessed book ratings data 14
3.10 Preprocessed Emotion Dataset 15

4.1 Model analysis for Summarization 18

5.1 BERT Classification Report 30
5.2 RoBERTa Classification Report 31
5.3 DistilBERT Classification Report 31
5.4 Evaluation Metrics for Different Models 33
5.5 Results for book recommendation 35
5.6 Performance Metrics of Sentiment Analysis 37

Nomenclature

The next list describes several symbols & abbreviation that will be later used within the body of the document

ϵ	Epsilon
υ	Upsilon
$BART$	Bidirectional and Auto-Regressive Transformers
$BERT$	Bidirectional Encoder Representations from Transformers
$BLEU$	Bilingual Evaluation Understudy
$DistilBERT$	Distilled version of BERT
$FAIR$	Facebook AI Research
GPT	Generative Pre-trained Transformer
GSG	Gap Sentences Generation
kNN	K-Nearest Neighbor
MAE	Mean Absolute Error
MLM	Masked Language Model
$MultinomialNB$	Multinomial Naive Bayes
NLP	Natural Language Processing
$PEGASUS$	Pre-training with Extracted Gap-sentences for Abstractive Summarization
$RMSE$	Root mean Squared error
$RoBERTa$	Robustly Optimized BERT Approach
$ROUGE$	Recall-Oriented Understudy for Gisting Evaluation
SVD	Singular Value Decomposition
$T5$	Text-To-Text Transfer Transformer
$TF-IDF$	Term Frequency - Inverse Document Frequenc

Chapter 1

Introduction

1.1 Background and Motivation

In the current times, Artificial Intelligence (AI) and Machine Learning (ML) both have greatly emerged and are impacting various industries along with all sorts of sectors, playing a significant role. AI refers to the involvement of programming computers to perform tasks which need human-like intelligence. While on the other hand, ML mainly focuses on the creation and development of algorithms that enable computers to learn from examples and improve decision-making continuously based on datasets. The combination and integration of AI and ML has not only changed industries but has also brought advancements in many fields, including libraries, improving the overall service along with an enhanced user experience. Even though libraries are all about books and papers, bringing technology into the mix allows them to handle a ton of data and gather useful insights. At present, there is already an abundance of information and data in the libraries which are filled with books, papers, magazines, and other forms of writing. Finding any particular topic from thousands of these books is not an easy task. Even if we are able to find related topics manually, there is a chance to miss out on the huge other papers that may exist. And this is where the integration of AI, ML, and NLP can help us out in reshaping how traditional libraries work and turning them into smart libraries containing a helping hand as a bot and an assistant. Here, ML algorithms step in to handle routine tasks like sorting data, making recommendations, and categorizing things, freeing up librarians from jobs that require a specialized expertise, and reducing the need of any complex or dull workload. This paper sets out with an aim to give you a good look at how AI and ML can come into play in library systems. The introduction of an advanced library assistant system has been explored here taking the help of AI, ML, and NLP to revolutionize knowledge management. The collected data have been combined to form a proper analysis of the library, giving it a thorough overview to spot the common techniques used in the library world. By putting these systems to work, libraries can not only become more versatile but also remove the natural hiccups in human judgment, ultimately reducing the possibilities of errors. Even though AI and ML systems outshine librarians in terms of results and accuracy, it's crucial to acknowledge the unique role librarians play. Librarians are able to express feelings and make ethical calls, which AI systems are yet unable to perform. Rather than phasing librarians out, integrating AI and ML simplifies the complex, time-consuming tasks, which in the end makes library operations smoother and more efficient for both the staff and visitors. In summary, embracing AI and ML aligns with the wider trends in the digital world and does not mean any danger or trouble for the librarians, instead works as a helping resource. But to make this happen in the proper way, access to big datasets, powerful computing, and the development of open-source ML frameworks is necessary. Techniques like Natural Language Processing (NLP), image recognition, and sentiment analysis give libraries the tools to pull insights from messy data, automate metadata creation, and boost information discovery. The combined use of AI and ML opens up a massive opportunity for libraries to upgrade their services and manage operations more effectively.

1.2 Research Problem

Libraries are a crucial establishment of any society. Not only are they vital to our knowledge, but also a house of history. Having a large significance to the society, the challenges are also overwhelming for any traditional libraries at this age of time.

- With the increase of books, journals, articles, and papers, the information that has to be handled by a library has also increased significantly. Traditional libraries often lack the ability to handle all of them in an efficient manner. Even at times, the libraries have to get rid of older materials, just to be able to make room for the newer books. But this should not be the only solution, especially at a time when we are at a blooming stage of technology.

- There also remains the nonexistence of any personalized user guidance. As we are living in the modern times of technology, all our data is already available at hand on our mobile phones, and so the ability to get a personalized user experience should also be just as seamless. Additionally, it is important to understand the user's sentiment while interacting. Through this, it should become easier to understand how the user reacts to specifics and this will in return be able to adapt to those preferences.

- Although there may be different recommendation systems that exist, that are able to recommend books to the users according to their preferences, as well as different summarization systems, there remains the need for having them combined at one place. Having both of these systems at one place for any library would make it more easier and fun to choose books that the readers would enjoy to read. And to add to this, with the advancement of modern technology, where everything is becoming easier to be accomplished, having something that will also be able to read out parts from the book is necessary. This will simply enable the ability to multitask and make life more simple and convenient.

- The number of people visiting the libraries is decreasing day by day. This is a concerning issue, as reading books and utilizing the libraries for personal development and knowledge gathering hold a significant importance. Without these practices, the nation as a whole may simply not be able to thrive in the challenging world that demands having a diverse knowledge.

- There also remains the challenge of navigating large libraries and finding the sections. This may not be an issue for someone who regularly visits the library, but for an average person, this poses a challenge to really find the written materials that they would truly be interested in. Also, different libraries may use different techniques to arrange their books, which just makes it complex to truly get used to being able to simply just go and locate the books that one intends to get.

1.3 Research Objective

In the pursuit of transforming a traditional library to be able to handle present challenges to modernize them, a direct assessment of the system is necessary. Our research aims at establishing an overall system for the combination of different systems with an assistant and robot system that will act as a helping companion, resulting in a convenient and stress-free experience when visiting the library. For being able to achieve this, the goals that we mainly have targeted have been elaborated on below.

- The primary objective here is the implementation of a smart system that would help in the creation of a proper library assistant system. For doing so, some of the necessities are for the assistant to be able to respond to the user to questionnaires regarding the library, like asking regarding what time the library starts operating from.
- Additionally, it would be able to readout books to the user by using a text-to-speech system. Alongside this, the assistive system would also be able to summarize and keep track of chapters if necessary for the user which would help create a seamless user experience.
- The assistant system should also be able to analyze the sentiment of the user, by understanding their sentiment as they provide feedback to each response. By doing so, a tailored recommendation system can also be implemented which can result in a more engaging and relevant user experience.
- As there is not much research regarding libraries sector especially in Bangladesh, by working on this system, we will be able to gather numerous datasets which will be crucial for the entire workflow as well if necessary for being used in future workings. For instance, collecting and gathering numerous Q/A related to libraries which is crucial data for training the assistant. Besides, an overall dataset is also required containing the contents of different books along with their chapters and summaries. This can be used for being used to read out from anywhere of the book to the user and also give a precise idea regarding the book. There is also the necessity of a dataset containing the genre of books and their ratings that need to be collected from actual users. This then can be used for the recommendation system. In short, a list of diverse datasets are required to be able to properly train and make use of the library assistant system.
- After doing these, we will be able to collect the overall success rate of the system regarding how reliable it is and how efficiently it is working as a whole.

For this, metrics such as reduced search time, user engagement, and any improvement in sentiment towards the library services can be taken into account.

- Regarding navigation, the proper use of a robot system is necessary that will be able to navigate throughout the library and assist the users to find the section of the library that they are looking for. This should be able to eliminate the need of a complex map or even a complex code for each section/part of the library. This system shall also be able to provide intuitive guidance, aiding disabled users to easily understand and navigate through the library.
  Ultimately, the objective is to create an interactive and user-centric experience, bridging the gap between traditional and modern library services. The research objectives collectively form a structured framework towards the development of a smart library system, which will not only address the current challenges and issues but also anticipate the evolving need for an intelligent library in the digital age.

Chapter 2

Related Work

The literature study delves into the dynamic interaction of Artificial Intelligence (AI), Machine Learning (ML), Natural Language Processing (NLP), and library management, taking readers on an intellectual journey via a wide range of academic papers and publications.

2.1 Analysis of Qualities Needed for AI Machines in Libraries

In a journal article, Barki (2022) talks about the qualities needed for AI machines include qualities like Representation, Search, Reasoning, and Knowledge Acquisition. These are what is necessary for the development of an NLP-based AI robot for smart interaction and support in physical libraries. Also by doing so, it creates an immersive experience in libraries, with the aim of enhancing user engagement that aligns with the idea of the proposed NLP-based AI robot [18]. In addition, the study also analyzes how with the help of robotics visitors can be assisted and guided in libraries, to find whatever they are looking for more quickly and solve their general queries right away instead of waiting for an employee to respond. The study of these technologies is acknowledged for providing a framework for evaluating the benefits that this offers in addition to the challenges of creating an NLP-based AI robot [18]. Similar to our situation, these insights can be applied to evaluate the effectiveness and implications of related technology for modernizing libraries.

2.2 Traditional Library Management Systems

In a paper, Iwayemi et al. (2019) critically examine the challenges that come attached to traditional library management systems and in response propose a software-based solution to address these issues [8]. The key challenges identified include data loss, time consumption, and the risk of losing materials/books in a traditional library system. This aligns with the efficiency objectives stated in relation to NLP-based AI robots for physical library help and intelligent engagement. The paper also highlights the drawbacks in previous works that are related to library management systems, for instance, the issues with cost-effectiveness, platform dependency, and inadequate infrastructure maintenance [8]. This information offers us great insights as it offers an overall knowledge about the challenges that traditional library systems face which reinforces the need for innovative solutions. Here also the proposed solution is given as creating a robust library management system using HTML, CSS, and PHP for the front end, and MySQL for the back end. Then by following a waterfall model for the software system, success can be achieved in managing and maintaining an accurate database. However, it is noted that in this implementation there lacks the integration of artificial intelligence or complete automation of the process, keeping the incompleteness of an intelligent library management system.

2.3 Analysis of Automation in Libraries with AI

The research by Prof. Shailendra Vidhate provides a comprehensive analysis of the status of automation in libraries with a focus on the application of AI [11]. The main idea here is that AI will make a big difference in how effective library management systems are along with their intelligence. This coincides with the main objective of the NLP-based AI robot, which is to facilitate human-like interaction in physical libraries. The research presents applications based upon AI as possible solutions in libraries by exploring concepts that are linked to robot-based implementations. This aims at improving procedures like issuing and receiving books [11]. Furthermore, the proposal of a model that includes entities including librarians, students, users, and a database, emphasizing a streamlined user experience marked by speed, security, and easy access has been made here [11]. The complexity of the design, the limitations of natural language processing, and the awareness of the whole knowledge barrier have also been discussed here. These insights provide valuable considerations for the implementation of NLP-based AI robots, which in turn helps in anticipating the potential challenges for obtaining optimal performance.

2.4 Integration of AI for Efficient Library Management

In Kaijun Yu's article integration of artificial intelligence to establish an intelligent system for more efficient and effective library management [12]. Here various existing technologies that can contribute to the management of library resources, such as RFID systems and bookbots utilizing high-density automated shelf technology have been discussed [12]. These technologies connect particularly well for the implementation of a smart library, especially in terms of simplifying book organization and retrieval. Additionally, the mention of the implementation of CAPM in Johns Hopkins University Library adds a practical dimension to the incorporation of AI in real-world library environments. There still remains a challenge to finding a solution for reducing lost books through the implementation of AI and robotics [12]. Aside from these, the article also suggests the adoption of intelligent applications, consulting services, and inventory management systems to significantly improve the overall user experience in libraries that can be both beneficial for the visitors and employees [12].

2.5 Potential Evolution of Smart Libraries

The research paper explores the potential evolution of smart libraries driven by advancements in AI. Here the researchers envisioned a future where Intelligent Systems (IS) and Expert Systems (ES) powered by AI could potentially replace traditional librarian roles [4]. The paper places significant importance on the use of Natural Language Processing (NLP) and robust systems in replicating librarian tasks, which ultimately can create a smoother experience for the customers and become a more accessible library experience for the readers [4]. The study embarks on the important claim that AI can be trained to become as intelligent as humans in some scenarios by using efficient algorithms and appropriate data management. [4]. Furthermore, the integration of Expert Systems (ES) and Intelligent Systems (IS) is highlighted here, along with ES incorporating knowledge from multiple librarians to produce refined and unbiased results [4]. This approach closely lines up with the NLP-based AI robot's intention to integrate various AI techniques for a comprehensive library support system. In this research there is also a survey taken among librarians, collecting opinions on questions related to library functions. The data collected contributes to the identification of four AI criteria that are categorized as Expert System (ES), Recommender System (RS), Natural Language Processing (NLP), and Data Mining (DM). These categories are then used to enhance the Iran Library system in three areas, namely: Public Services (PS), Technical Services (TS), and Management Services (MS) [4].

2.6 Emulation of Expert System for Library Support

In a Journal Divayana et al. (2015) concentrates on utilizing methods such as Forward Chaining and Backward Chaining to emulate an Expert System instead of a traditional librarian [3]. Forward Chaining, which is a data-driven method that continuously incorporates new information based on user needs and runs various rules to make decisions. Inversely, Backward Chaining is a goal-driven method where the system starts with a goal, then creates rules according to received goals, and finally provides decisions based on user preferences. These approaches are similar to the Expert System as mentioned regarding Forward Chaining and Backward Chaining [3]. This context-specific expert system is supposed to be working like a librarian. With the help of techniques and algorithms to provide better outcomes than a single librarian this system is able to generate an overall good response [3]. The primary goal here was to expedite the library process and enhance accessibility for readers, which in turn ultimately would save time and improve the overall library experience.

2.7 Essential Techniques for Book Recommendations

According to Berbatova (2019) an overall summation regarding valuable insights into essential techniques for book recommendations, focus on content-based filtering, collaborative filtering, and recommender systems has been discussed [6]. Content-based filtering here involves the content recommendations based on the user's previous questions and interests. It might suggest information that is comparable to what they have already searched about by using their past. This is close to the content recommendation aspect of NLP-based AI robots, where user interactions and preferences play an important role in tailoring suggestions to be more suitable for the user. Collaborative filtering, recommends content based on the preferences of users with similar interests or outputting a proper response. By the process of collecting and processing data taking from all the users, the system is able to identify and provide matching preferences and suggestions based on what others with similar interests have already given a good feedback [6]. This collaborative approach mimics the collaborative filtering component of the recommendation system in the proposed NLP-based AI robot, which overall enhances the personalization of suggestions for the end users. In addition the recommending systems, as described in the paper, are algorithms that process collected data to offer personalized recommendations to users, leveraging a combination of content and collaborative filtering to enhance the recommendation process [6].

Chapter 3

Dataset and Preprocessing

This section will cover the methods utilized to acquire the dataset for the research as well as which dataset was used in this particular case.

3.1 Dataset Collection and Creation

3.1.1 Query to function Dataset

We manually created a unique dataset in order to improve our chatbot's abilities. We train models to make the chatbot learn what customers desire from this dataset. This was accomplished by surveying people to see how they ask for things and collecting a bunch of different ways people phrase their questions. It's a detailed dataset that shows the ins and outs of how people talk, setting up a solid base for our chatbot to figure out what users are after. For simplicity purposes, we only kept two rows in the dataset. Those are:

Queries	**Function**
Are there any library-led workshops on information literacy?	others
Articulate the summary of "The Three Musketeers" by Alexandre Dumas.	tts
Briefly explain the central concepts covered in the last chapter of "1984."	summary
Can you recommend a book like 'Airframe'?	recommendation

Table 3.1: Example of the Dataset for Chatbot

- **Query:** There are around five hundred different queries collected from various sources, each carefully labeled with the appropriate function the query requests.
- **Function:** Four different categories: tts, recommendation, summary, and others are used to classify the the queries. A brief sample of several queries with various functions may be seen in table 3.1. Figure 3.1 illustrates the variety of functions included in the dataset. The richness of this dataset makes it possible to better understand user queries and facilitates the training of the model for precise, context-aware function prediction.

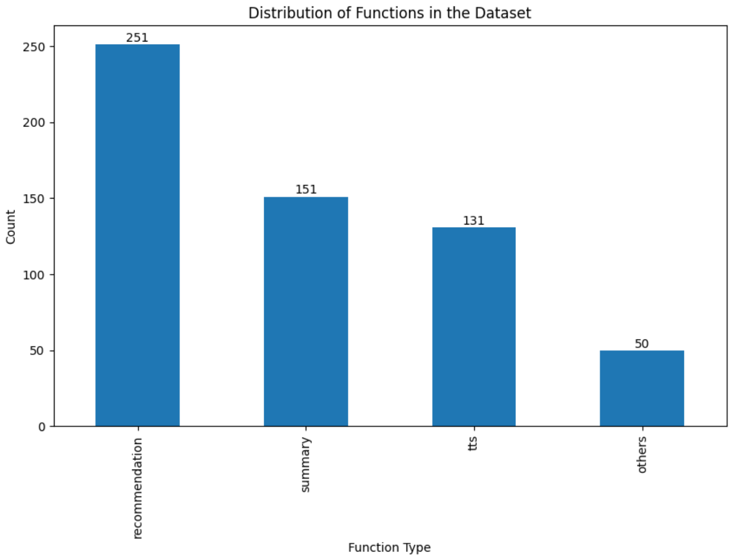


Figure 3.1: Variation of functions in the Chatbot Dataset

3.1.2 Library Question and Answer

The library chatbot’s Intent Dataset is carefully selected, and derived from a variety of sources to guarantee inclusiveness and thoroughness when expressing user inquiries. Using a multimodal approach that combines survey data, web scraping from multiple online sources, and extraction from library FAQ sections, the dataset captures a range of conversational elements, such as general hellos and goodbyes, common questions about libraries, and common inquiries made to librarians. From all the available categories, three important categories in terms of libraries are shown in table 3.2, table 3.3, and table 3.4. Quality assurance procedures have been implemented to improve the dataset and eliminate any biases and redundancy. Additionally, the dataset is made to be updated dynamically, which allows it to adapt to changes in user patterns, library services, and language usage over time and ensures its relevancy. The involvement of users has been essential, as seen by feedback loops and user surveys that help modify datasets according to real user demands. The intent dataset offers possible use cases outside chatbot creation, such as analytics and insights into user behavior in library-related interactions, in addition to improving the chatbot’s natural language comprehension skills.

Patterns	**Responses**
”What is the cost of ’{book_name}’?”	”The cost of ’{book_name}’ is ${price}.”
”How much does ’{book_name}’ cost?”}	”For ’{book_name}’, the price is ${price}.”
”Tell me the price of ’{book_name}’.”}	”’{book_name}’ is priced at ${price}.”

Table 3.2: Category - Cost of Books

Patterns	Responses
"Where can I find '{book_genre'?"}	"You can find '{book_genre}' in the {section} section."
"In which section is '{book_genre' located?"}	"The section for '{book_genre}' is {section}."
"Can you guide me to the section for '{book_genre'?"}	"'{book_genre}' is located in the {section} section."

Table 3.3: Category - Library Section

Patterns	Responses
"What facilities do you provide?"	"Experience a welcoming environment with study rooms, online catalog access, and a cozy reading corner. Visit the library from 9:00 AM to 5:00 PM, Sunday to Thursday."
"When does the library open?"	"Explore our diverse book collection and amenities including reading areas, and quiet study spaces. Enjoy free Wi-Fi, printing. You can find us Open from 9:00 AM to 5:00 PM, Sunday to Thursday."
"What are the library hours?"	"Our library offers comfortable reading areas, computer access, printing services, and a quiet study environment. We are open from 9:00 AM to 5:00 PM, Sunday to Thursday."

Table 3.4: Category - Library information

3.1.3 Books

A manually created dataset, including book titles, author names, table of contents, summary, and individual chapters, was used in this research. Data entry was required for the dataset creation, and books that are freely accessible on the internet were chosen without regard to any particular criteria. To be precise, eBooks that we discovered on Project Gutenberg[19]. In order to guarantee that a variety of information was included and to make the assessment of summarization models easier across genres and themes, a manual curation strategy was selected. An illustration can be seen in Table 3.5.

Title	Author	Contents	Summary	Chapter-1	Chapter-2	Chapter-N
Rich D...	Rober...	Intro...	Two dads...	Havin...	The poor...	It's not...
Lesso...	Edit...	Chapt...	Lessons...	It is...	It has...	We must...

Table 3.5: Ebook available for Summary and TTS

3.1.4 Collaborative Book Recommendation System

For our book recommendation system, we have used 3 types of datasets. All these data are obtained from kaggle[13]. Since we used collaborative filtering, we need data about the users and books, along with the data about the ratings of different books by different users.

ISBN	Book-Title	Book-Author	Year-Of-Publication	Genre
345402871	Airframe	Michael Crichton	1997	Thriller

Table 3.6: Book Information

userID	ISBN	bookRating
277427	002542730X	10
3363	002542730X	0
12538	002542730X	10
13552	002542730X	0
16795	002542730X	0

Table 3.7: Ratings

- **User Information:** Consists of "userID" for each user.
- **Books Information:** Includes "ISBN", "bookTitle", and "bookAuthor" for each book. A visual representation can be seen in table 3.6.
- **Ratings:** In table 3.7 we see that it contains "userID", "ISBN", and "bookRating" for books that have been rated, along with corresponding user and book IDs.

3.1.5 Emotion Dataset

The Emotion Dataset[21] included in this study was sourced from Kaggle, a well-known machine-learning dataset platform. The two main rows in the dataset are "Comment" and "Emotion."

Comment	Emotion
i seriously hate one subject to death but now...	fear
im so full of life i feel appalled	anger
i sit here to write i start to dig out my feel...	fear
ive been really angry with r and i feel like a...	joy
i feel suspicious if there is no one outside li...	fear

Table 3.8: Emotion Dataset

- **Comment:** There are around six thousand distinct phrases, each carefully labeled with the appropriate emotion.

- **Emotion:** Three different categories: joy, fear, and anger are used to classify the emotions recorded in the dataset. Table 3.8 contains an illustration of the data. We are able to study and analyze the intricate relationships that exist in these specific emotional domains between language and emotion thanks to this dataset. The range of emotions found in the dataset is shown in the figure 3.3. Because of the richness of this dataset, sentiment can be understood more deeply and the model can be trained more easily for accurate, context-aware sentiment analysis.

3.2 Preprocessing

3.2.1 Query to function Dataset

"Downsampling" has been used to produce a balanced dataset. Initially, the dataset has varying numbers of queries for every function. Our goal is to make sure that there is an equal number of queries for each function since an imbalanced dataset might lead to bias toward a certain function. This would help to prevent bias during the training of machine learning models. In comparison to the earlier data in figure 3.1, we can observe the number of requests for each function in 3.2 after the downsampling.

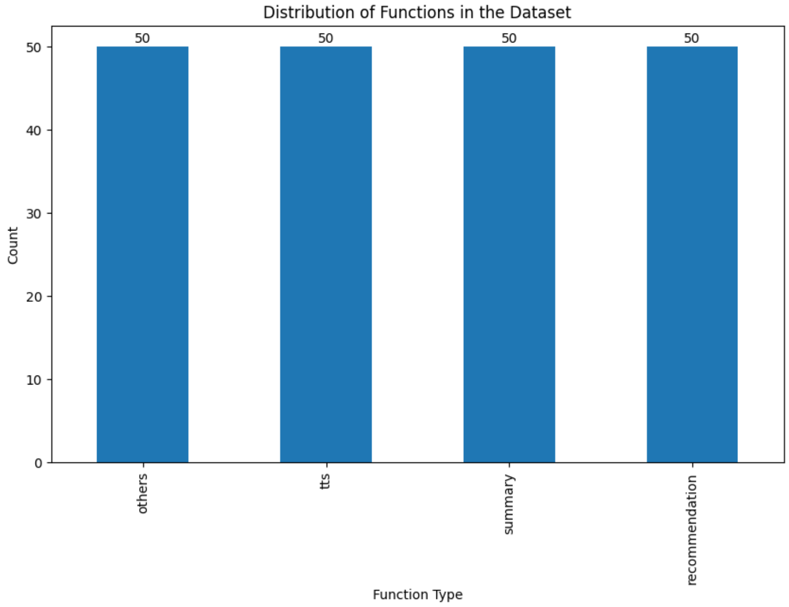


Figure 3.2: Variation of functions in Chatbot Dataset after preprocessing

3.2.2 Books

Preprocessing or encoding isn't necessary because the dataset was chosen manually. Still this dataset must undergo processing for the particular model that will be used because it was chosen with the intention of summarizing a book's contents. The dataset, consisting of chapters from various books, underwent a series of steps to ensure compatibility with the Pegasus model and its associated tokenizer.

- **Hugging Face Transformers Toolkit:** With its extensive library of tokenization tools and pre-trained models, the Hugging Face Transformers library proved invaluable in managing the intricacies of natural language processing jobs. Models like Pegasus might be easily included in the dataset preparation workflow by using this toolbox.

- **Pegasus Tokenizer:** This tokenizer automatically divides textual information into smaller parts called as tokens. It was created especially for the Pegasus model. To ensure that the input is in a format that is consistent with the architecture of the model, the tokenization procedure include managing subword units and special tokens in addition to breaking the text into words.

  Using Hugging Face Transformers' PegasusTokenizer allowed for uniform and effective tokenization throughout the dataset. To train the Pegasus model to provide logical and instructive summaries, this consistent representation of textual input was essential.

3.2.3 Collaborative Book Recommendation System

For pre-processing the recommendation system dataset, we need to perform data cleaning and data merging, also turn the data into a matrix for implementing kNN and SVD.

bookTitle	**totalRatingCount**
Politically Correct Bedtime Stories: Modern Ta...	82
Pride and Prejudice	64
Echoes	54
Fahrenheit 451	119
Me Talk Pretty One Day	146

Table 3.9: Preprocessed book ratings data

Firstly, we got rid of all the unnecessary columns that would not have any influence over our system. For example, we got rid of columns from the book information (Table 3.6) like 'yearOfPublication', 'publisher'. After that we only selected users that have a rating higher than 200 and books that have a rating higher than 100. These are to filter out users that have not participated in rating a good enough amount of books. After that the book information and ratings are merged to one data, like shown in table 3.9. Then the books that cross the 50 threshold of rating are considered as popular. Now, with kNN and SVD it would be simple to build a matrix and run the algorithm to find us our desired outputs.

3.2.4 Emotion Dataset

We will do a number of preprocessing steps, including tokenization, stop word removal, stemming, and lemmatization, to ensure the dataset is of high quality. Simply putting the data in a format that allows for analysis and prediction is known as preprocessing input text. If appropriate, data augmentation techniques are also looked at in an effort to broaden the dataset's diversity. Here's a thorough breakdown of some of the techniques used to clean the dataset.

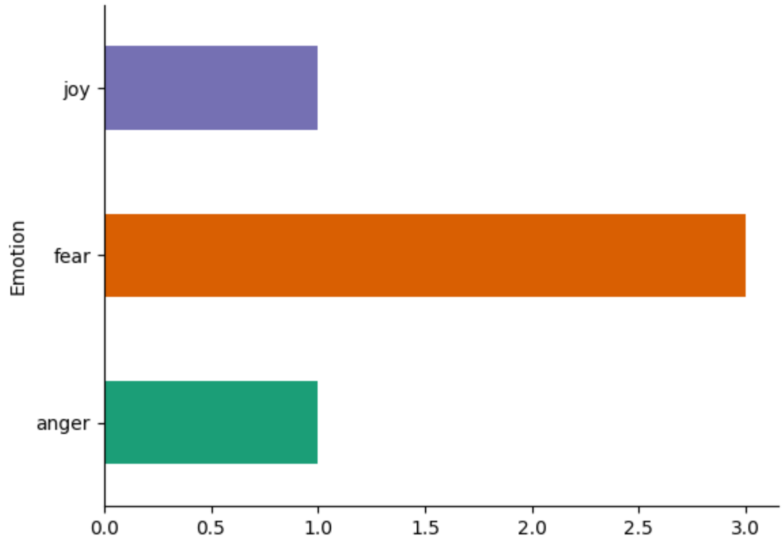


Figure 3.3: Emotion Dataset Chart

Comment	**Emotion**
im so full of life i feel appalled	negative
ive been really angry with r and i feel like a...	positive
i feel jealous becasue i wanted that kind of l...	negative
when a friend of mine keeps telling me morbid ...	negative
i finally fell asleep feeling angry useless an...	negative

Table 3.10: Preprocessed Emotion Dataset

- **Tokenization:** Text preparation is the initial step in every NLP project, according to Menzli[15]. It is the procedure for dissecting the text corpus into its component parts. These distinct components serve as input for algorithms used in machine learning. For example the sentence ”I'm so full of life I feel appalled.” will be converted into:

  {*I'm, so, full, of, life, I, feel, appalled.*}

- **Stop Words:** Stop words, which include ”a,” ”the,” ”is,” and ”are,” are a group of words that are often employed in a language. Stop words are essential for weeding out words that provide little or no useful information in natural language processing (NLP). Stop words are essential for increasing the effectiveness of language processing algorithms and lowering noise in textual data, as stated by Ganesan in 2019[7].

- **Stemming and Lemmatization:** Porter Stemmer's approach of stemming involves carefully removing frequent prefixes and suffixes from words to reduce them to their most basic components. This condensing results in a more straightforward representation in addition to aiding in the reduction of words to their most fundamental forms. Stemming is crucial in mitigating the challenges posed by word variations, hence improving the efficacy of subsequent natural language processing tasks. Lemmatization, on the other hand, maps words to their dictionary or base forms to provide further information on the linguistic structure of words. This entails taking into consideration elements like contextual significance and grammatical accuracy. Lemmatization and stemming are crucial steps in natural language processing that reduce words to their most basic forms. Several processes help in language normalization, according to Cambridge University Press (2009)[2]. The Porter Stemmer is a well-known technique that creates grammatically sound base forms by lemmatization. For example -

$$\text{"running"} \rightarrow \text{"run"}, \text{"jumps"} \rightarrow \text{"jump"}, \text{"happily"} \rightarrow \text{"happily"}$$

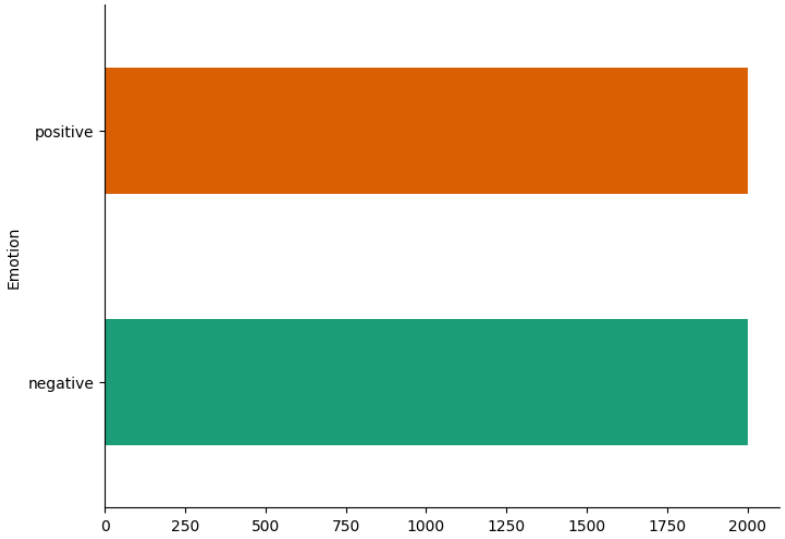


Figure 3.4: Preprocessed Emotion Dataset Chart

After the preprocessing and cleaning are done, the new state of the dataset in table 3.10 can be observed. Also the new variation of the emotion in figure 3.4.

Chapter 4

Proposed Methodology

4.1 Background

The proposed approach incorporates several machine-learning techniques to improve user experience and engagement in the literary domain, building upon collaborative filtering for book suggestions. Personalized book suggestions are generated by utilizing kNN (k-Nearest Neighbors) to enable collaborative filtering and find people with similar tastes. Beyond this, Multinomial Naive Bayes (MultinomialNB) and TF-IDF are used in sentiment analysis to provide a deeper comprehension of users' emotional reactions to literature.

The approach also includes abstractive summarization with Pegasus, which simplifies lengthy textual information for effective understanding and extracts important ideas. In order to improve accessibility even more, chapter text may be converted into audio material through the use of Google Text-to-Speech (GoogleTTS) technology, providing a different way for people with different tastes to interact.

4.1.1 Transformer based Models

For Chatbot

Bidirectional Encoder Representations from Transformers (BERT)

BERT stands for Bidirectional Encoder Representations from Transformers, which was developed by Google. It is a pre-trained model that is used for natural language processing(NLP). BERT is one of the transformer architecture models that is especially designed for understanding the context and meaning of a language. BERT is widely used for query, intent classification processes. BERT's ability to understand contextual information and find the relationship between words makes it a good model for understanding natural language.

Robustly optimized BERT approach (RoBERTa)

RoBERTa stands for Robustly optimized BERT approach. This model runs on top of BERT with some modifications. This model was developed by Facebook AI to make a better version of BERT. The working principles of RoBERTa and BERT have similarities, but there are some modifications and improvements that

makes RoBERTa better at identifying natural language tasks, which includes query classification.
Unlike BERT, RoBERTa uses Dynamically Masked Language Model (MLM) during the pre-training process. In MLM, instead of using the same masked pattern for every example, RoBERTa randomly picks batches with different masking patterns. Secondly, RoBERTa is trained in Larger corpus in comparison to BERT, making it more accurate in classification. Thirdly, in RoBERTa Next Sentence Prediction (NSP) is being removed, which was initially used in BERT, this change has caused the model's understanding to be more effective.

Distilled verion of BERT (DistilBERT)

DistilBERT is the distilled version of BERT (Bidirectional Encoder Representations from Transformers), which was introduced by Hugging face. The main goal of this model is to design a computationally efficient variant of BERT. BERT being a computationally heavy model, the goal of DistilBERT is to retain the performance of BERT with less computational power. DistilBERT accomplishes this by changing the parameters and integrating small modifications to BERT.
DistilBERT uses distillation knowledge during the training process. Here, the pre-trained BERT functions as a teacher to the DistilBERT, and it is trained to mimic the output of BERT with less time and resources. DistilBERT does this by using lesser parameters during training. Also, Next Sentence Prediction (NSP) is removed in DistilBERT to make it lighter than BERT.

For Summary

Model	Model Name	Tokenizer	Model Architecture
BART	facebook/bart-large-cnn	BartTokenizer	BartForConditionalGeneration
PEGASUS	google/pegasus-cnn_dailymail	PegasusTokenizer	PegasusForConditionalGeneration
T5	t5-small	T5Tokenizer	T5ForConditionalGeneration

Table 4.1: Model analysis for Summarization

Pre-training with Extracted Gap-sentences for Abstractive Summarization (PEGASUS)

PEGASUS implements the seq2seq architecture in the same manner as any other sequence transduction job. But this architecture's self-supervised pre-training goal is what makes it unique. PEGASUS is an abstractive summarization model that is transformer-based. It makes use of a unique self-supervised pre-training goal known as gap-sentences generation (GSG), which is intended to achieve high performance on downstream tasks linked to summarization. An explanation is shown in figure 4.1 where we see the working architecture of the model. Similar to an extractive summary, PEGASUS groups together the most significant sentences from an input text and generates an output sequence out of the remaining sentences. Pre-training the encoder as a masked language model makes sense because PEGASUS's basic architecture comprises of an encoder and a decoder, despite the GSG being its primary contribution. By using other words from the sequence to anticipate the words that are randomly masked from the sequences. The GSG job is based on this idea and may be understood as a document-level MLM.

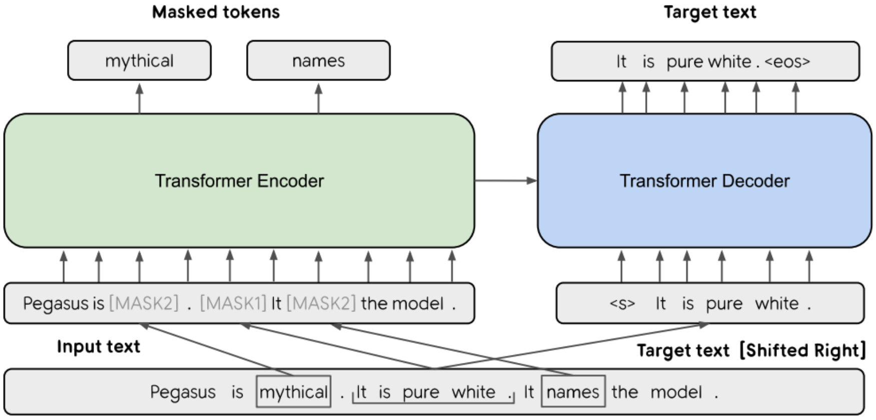

Masked tokens
Target text
mythical
names
It is pure white . <eos>
Transformer Encoder
Transformer Decoder
Pegasus is [MASK2] . [MASK1] It [MASK2] the model .
<s> It is pure white .
Input text
Target text [Shifted Right]
Pegasus is mythical . It is pure white . It names the model .

Figure 4.1: MLM (left) + GSG (right) training together in PEGASUS

According to the PEGASUS publication, this example uses both GSG and MLM concurrently as pre-training goals. Three sentences are present at first. Target generation text is one sentence that has been disguised with [MASK1] (GSG). The remaining two phrases are kept in the input, but [MASK2] (MLM) randomly masks certain words[23].

Bidirectional and Auto-Regressive Transformers (BART)

Facebook AI Research (FAIR) created the transformer-based BART, or Bidirectional and Auto-Regressive Transformers, neural network architecture for sequence-to-sequence tasks, with a particular emphasis on abstractive text summarization. The FAIR-introduced approach is known for its adaptability in applications involving natural language processing. BERT's design, which is similar to a combination of the bidirectional encoder from BERT and the left-to-right decoder from GPT, represents a thorough pre-training method. Distinguished by its bidirectional training, BART performs very well on comprehension tasks and shows improved efficacy via text creation fine-tuning. It routinely achieves state-of-the-art results in summarization, question answering, and abstractive discussion tasks, demonstrating its strong performance in these areas.
The BART(2019) study describes a pre-training technique that combines auto-regressive and bidirectional transformers. BART is a sequence-to-sequence autoencoder for denoising that may be applied to a variety of scenarios[9]. Pretraining is divided into two stages:

1. using an arbitrary noising function to distort the text, and
2. learning a sequence-to-sequence model to rebuild the original text

According to Vaswani et al. (2017), BART is designed using a universal sequence-to-sequence Transformer. For the base model, the encoder and decoder have six layers each, whereas the big model has twelve layers in each [5].

Text-To-Text Transfer Transformer (T5)

A cutting-edge pre-trained language model built on the transformer architecture is the transfer text-to-text transformer (T5). It uses a single, unified text-to-text framework that can handle any job involving natural language processing (NLP) and turns the input and output into texts in the target language [14]. We know that T5 may be readily scaled up by changing the number of parameters (from 60M to 11B), which allows it to obtain higher performance on a variety of NLP benchmarks, according to Wang Xie Du Hu (2023) [20]. To further capture complicated semantic linkages and long-range dependencies in natural language texts, T5 uses a full-attention method. Numerous Natural Language Processing (NLP) tasks, including question answering, sentiment analysis, text summarization, and machine translation, have been effectively tackled by T5 [14].

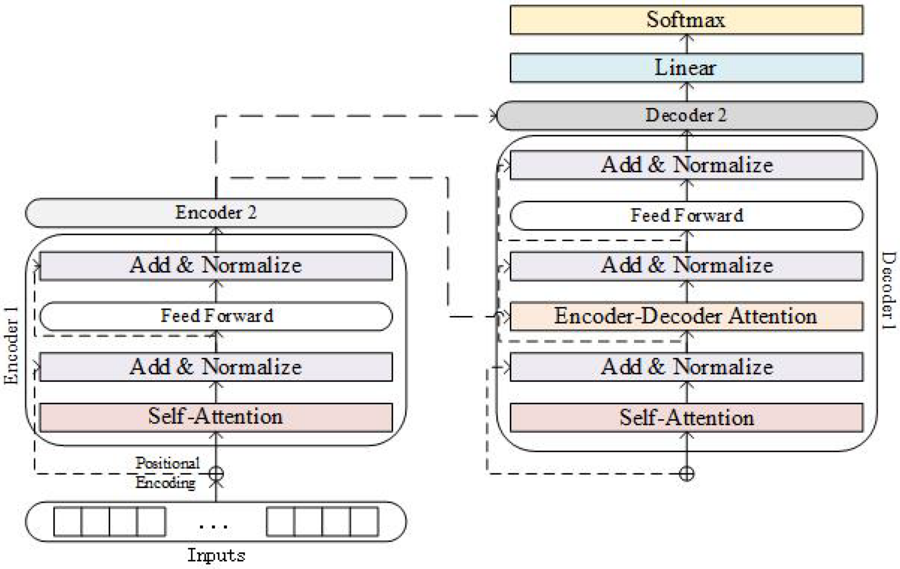


Figure 4.2: T5 Architecture

The T5 model follows the typical encoder–decoder structure, and its architecture is shown in figure 4.2.
Using distinct prefixes to denote distinct jobs is a fundamental aspect of T5's text-to-text architecture, which converts all NLP difficulties into text generation challenges. For instance, T5 may be used to do sentiment analysis on a given statement by simply prefixing it with ”sentiment:” and producing the words ”positive” or ”negative” as the result. With the help of this feature, it is possible to train a single model to handle several jobs without having to modify its architecture or goal function.

4.1.2 k-Nearest Neighbor(kNN)

K- Nearest Neighbour (k-NN) kNN is a classification algorithm, which classifies or predicts the grouping of an individual datapoint. This can be used for regression or classification problems. k-NN is a learning algorithm. In k-NN, the 'k' nearest neighbors are identified from a single data point; these neighbors are selected based on the training dataset's distance metrics. The euclidean distance is used to measure the distance of the nearest neighbors and Euclidean distance Formula is given in (4.1).

$$\text{Euclidean Distance}(D) = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \tag{4.1}$$

here,(x_1, y_1) is the coordinates of one point and (x_2, y_2) is the distance of the other point. By calculating the distance between all the points from a fixed point we get all the different distances. The points with the lowest distance from the initial point are considered as the k nearest neighbors. After finding the k nearest neighbors, kNN predicts by taking the labels for classification and values for regression. The selection of these kNNs affects the algorithm's results. Even Though, kNN is a relatively easier algorithm, it can be used for very complex purposes. With proper preprocessing and data optimization, kNN can be used to predict results [1].

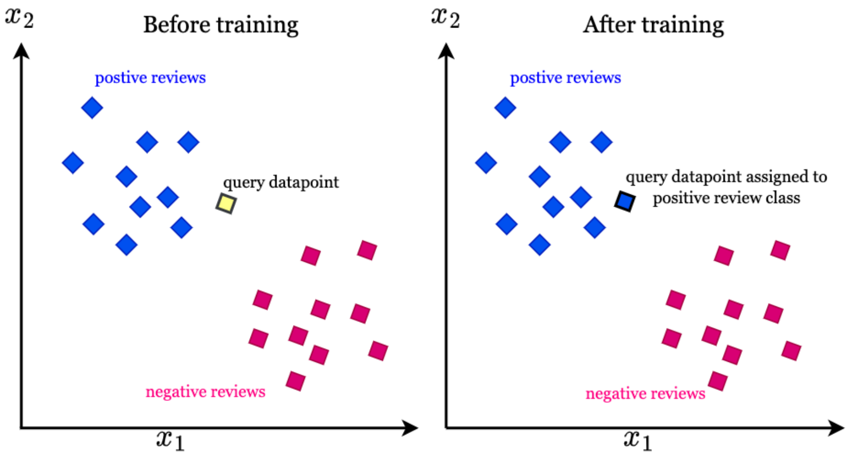


Figure 4.3: k Nearest Neighbor Algorithm

Architecture of k-NN

For using k-NN with the data, firstly we have created a user-item interaction matrix, with the filtered and processed data. Then the user-item matrix is converted into a sparse matrix. This is done to handle and store the sparse nature of the user item interactions. After that, the k-NN is trained on the sparse matrix, using the 'fit' method, the model is using the cosine distance metric and brute-force algorithm.

```
model_knn = NearestNeighbors(metric='cosine', algorithm='brute')
model_knn.fit(user_rating_matrix)
```

For demonstration purpose, a random book is selected for the system to recommend books.

```
query_index = np.random.choice(user_rating_pivot.shape[0])
distances, indices = model_knn.kneighbors(user_rating_
pivot.iloc[query_index, :].values.reshape(1, -1), n_neighbors=6)
```

With this distance and indices, the algorithm then identifies its nearest neighbors and recommends 5 nearest neighbors.

4.1.3 Singular Value Decomposition (SVD)

SVD is one of the matrix factorization techniques that is widely used in fields like linear algebra , signal processing, and machine learning. In SVD, matrix are decomposed into three other matrices, those are U, Σ and V^T. For a Matrix given as M the SVD shows it as $U\Sigma V^T$.

$$\text{SVD}(M) = U \cdot \Sigma \cdot V^T \tag{4.2}$$

Here, in equation (4.2), U is the $m \times n$ orthogonal matrix, Σ is the $m \times n$ diagonal matrix with singular values on the diagonal in descending order, V^T is the transpose of $n \times n$ orthogonal matrix V. As shown in figure 4.4, SVD is used mostly in dimensionality reduction, this is because it only allows one to approximate the orthogonal matrix by retaining only the most significant singular values. SVD is used for collaborative filtering in recommendation systems and also in latent semantic analysis in NLP. Uses of SVD are the most in the least squares problems, image compression. Recommendation systems, classification of handwritten digits, etc. [22].

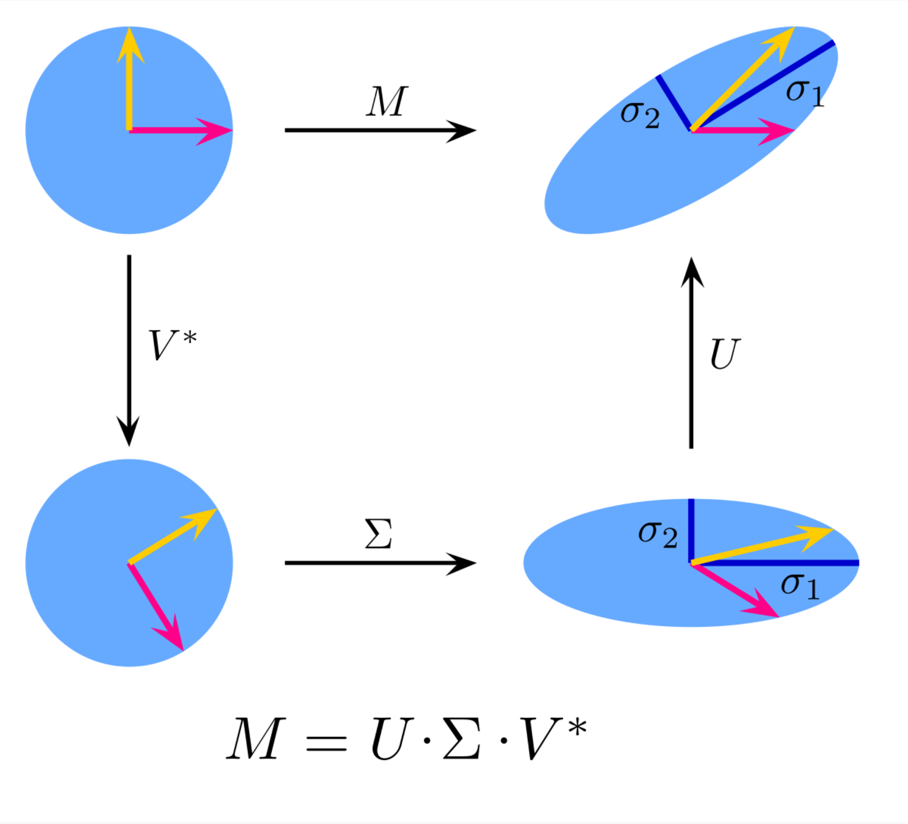


Figure 4.4: Singular Value Decomposition Algorithm

Architecture of SVD

In SVD also the user-item interaction matrix is used. Here, the SVD decomposes the user-item matrix into 3 matrices, U, Σ and V^T. This is the matrix factorization of SVD. What SVD is doing is, it is reducing the dimensionality of the original matrix into a lower number of latent factors. This is how SVD is used for dimensionality reduction. In the training phase, SVD learns the user-item factors by minimizing the difference between the predicted ratings, from those factorized matrices mentioned earlier.

```
model_svd.fit(trainset)
```

After training the SVD uses the trained factors to predict the missing ratings for the item that the user has not used yet.

```
predictions_svd = model_svd.test(testset)
```

4.1.4 Multinomial Naive Bayes (MultinomialNB)

The MultinomialNB model predicts the tag of a text, such as an email or a newspaper article, using the Bayes theorem given in equation (4.3).

$$P(A|B) = \frac{P(A) \cdot P(B|A)}{P(B)} \tag{4.3}$$

The probabilistic machine learning algorithm known as Multinomial Naive Bayes (MultinomialNB) is often applied to text classification tasks. It is best illustrated in Ray's journal, where the supervised learning algorithms known as ”naive Bayes” approaches are based on the ”naive” assumption of conditional independence between features, given the class variable [10]. These methods rely heavily on the Bayes theorem (Equation 4.3), which makes it possible to estimate class conditional probabilities effectively. In fact, MultinomialNB has shown to be effective despite the ”naive” assumption; this is especially true for text-related applications like document classification and spam filtering. The approach is straightforward since it estimates each distribution independently as a one-dimensional distribution; Ray's insights about the decoupling of class conditional feature distributions emphasize this. The algorithm's computational efficiency is enhanced by this decoupling, which makes it an appealing option in situations when speed is of the essence, particularly in contrast to more complex techniques.

4.1.5 Term Frequency - Inverse Document Frequency (TF-IDF)

We learn from Capital One that the statistical method known as TF-IDF is widely used in both information retrieval and natural language processing (NLP) [16]. It assesses a term's significance inside a document in relation to a group of documents. Text vectorization turns words in a document into numbers that represent their significance. TF-IDF is one of the most often used text vectorization scoring methods, however there are others. As the name implies, the TF-IDF method vectorizes or scores a word by multiplying its Term Frequency (TF) by the Inverse Document Frequency (IDF). A term's TF-IDF is determined by multiplying its IDF and TF values.

- **Term Frequency:** TF measures the frequency of a term (word) within a document using the equation (4.4). TF values are often normalized to prevent bias towards longer documents.

$$\text{TF} = \frac{\text{Number of times term } t \text{ appears in document } d}{\text{Total number of terms in document } d} \tag{4.4}$$

- **Inverse Document Frequency:** IDF evaluates the importance of a term across a collection of documents using equation (4.5). Adding 1 in the denominator avoids division by zero for terms that appear in all documents. IDF increases with the rarity of the term across the document collection.

$$\text{IDF} = \log\left(\frac{\text{Total number of documents in the corpus}}{\text{Number of documents containing term } t}\right) \tag{4.5}$$

- **TF-IDF:** TF-IDF combines TF and IDF to assign a weight to each term in a document relative to the entire document collection given in equation (4.6). The higher the TF-IDF score, the more important the term is to the specific document relative to the entire collection.

$$\text{TF-IDF} = \text{TF} \cdot \text{IDF} \tag{4.6}$$

The TF-IDF architecture serves as a fundamental basis for the extraction of relevant details from textual data, facilitating efficient document representation and supporting activities involving the comprehension of the significance of terms in context.

4.2 Robot Model

4.2.1 Robot system overview

A 3D-printed chassis with an aluminum structure for reinforcement and a mecanum drive system which provides diagonal movement which is necessary. 4 High torque motors are used for the drive system for movement in all possible directions and slops. Nvidia's Jetson Nano developer board was selected for image processing and to handle serial communication with a microcontroller (Arduino). Multiple generic cameras are used for image input. A single IMU MPU9250 is used for odometry which ensures proper path planning and execution. For obstacle avoidance and 3D surround mapping, a hobby-grade LIDAR sensor is used. As the robot is operating indoors a high-speed data transmission is achieved with a wireless 5Ghz adapter. A user can query the bot for information about any book, or destination and It will be able to communicate with the user about the information and fetch the book for the user. The bot uses machine vision (OpenCV, Yolo algorithm, and QR reader-python) for book and bookshelf detection (column detection).

1. High Torque Geared Motor, body, and Mecanum Wheels: A 3D-printed chassis with an aluminum extrusion profile as a base. Four high-torque motors with Mecanum Wheels are used for universal movement. A Rigid custom body with 350 RPM motors drives the whole system with BTS motor drivers (Change If needed).

2. Micro-Controller and Jeston Nano Developer board: Arduino Mega is used as a primary serial communication node which handles the serial data and plans the trajectory and path for the bot. The Serial command is sent through Nvidia's Jetson Nano which is used for image processing and obstacle avoidance. The powerful multicore processor is used instead of Raspberry pi 4 due to raw computing power. Raspberry Pi integration is possible.

3. Generic Webcams: Off-the-shelf webcams were used for image input. A resolution of 720p with 640x480 height and width was extracted for optimal model performance. The YoloV5 Image detection algorithm uses 640x640 as the default size. Multiple webcam is toggled for obstacle detection and avoidance.
4. Sensors: IMU, LIDAR, and with encoders are used to find the position of the bot in 3D space. Which is necessary for traversing through a Hallway or library. Slam is used for detecting distance and mapping.
5. Human Interfaces devices: User input is taken with a surround microphone to capture surround sound from a vicinity of 1 meter. A Display is used for visual stimuli. And Speakers are integrated for auditory stimuli.

4.2.2 Robot Architechture

NLP-based AI robot for smart interaction and on-site library support. A user (student) can query about any type of book to the robot (Virtual Librarian) and the onboard SoC will process the voice and convert it to a regular expression and some pointers and class to get a custom query package to the Library's database. The cameras are used for the detection and text recognition of the books and landmarks in the library. If the user wishes for the bot to fetch the desired book. The bot uses the onboard IMU for odometry and LIDAR for distance measurement and avoiding obstacles. Computer vision is used to aid with such a navigation stack.

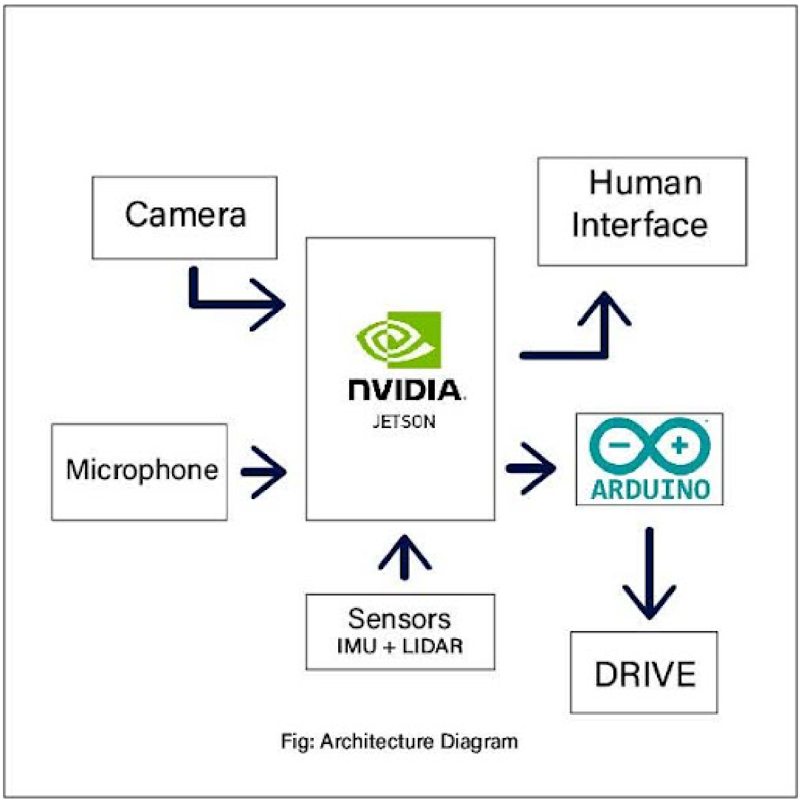


Figure 4.5: Robot Architecture

Here we assumed that visual landmarks will be there in the form of a QR code for the bot to recognize the bookshelves and so QR-reader a Python module is used in conjunction with OpenCV. Moreover, OpenCV and a model for text detection are used for detecting the book's text. YOLO is used for the detection of smoothing the path navigation process.

1. NLP and Visual Setup: Various Datasets and visual landmarks are labeled and classified to be used in the bot.

2. Hardware Requirements.: 3D printed Chassis, Aluminum extruded profile, Jetson nano, Arduino mega, Buck converter, 350 RPM High Torque motor, Mecanum wheel, Adafruit Motor shield, LIDAR, IMU MPU9250. 5Ghz/2.4Ghz adapter, Microphone, speaker, Camera, and screen. A picture in figure 4.6 shows all the used hardware for the robot system.

3. Circuit Diagram: The system is based on Jetson Nano with a Mega for controlling external movement. Here Arduino PWM pins are used for motor speed control. Encoders feed the Arduino with telemetry data. A buck module is used for powering the SoC. A direct 16V 4-cell Lipo battery is used for powering the whole system. And the whole architecture is visually explained in figure 4.5.

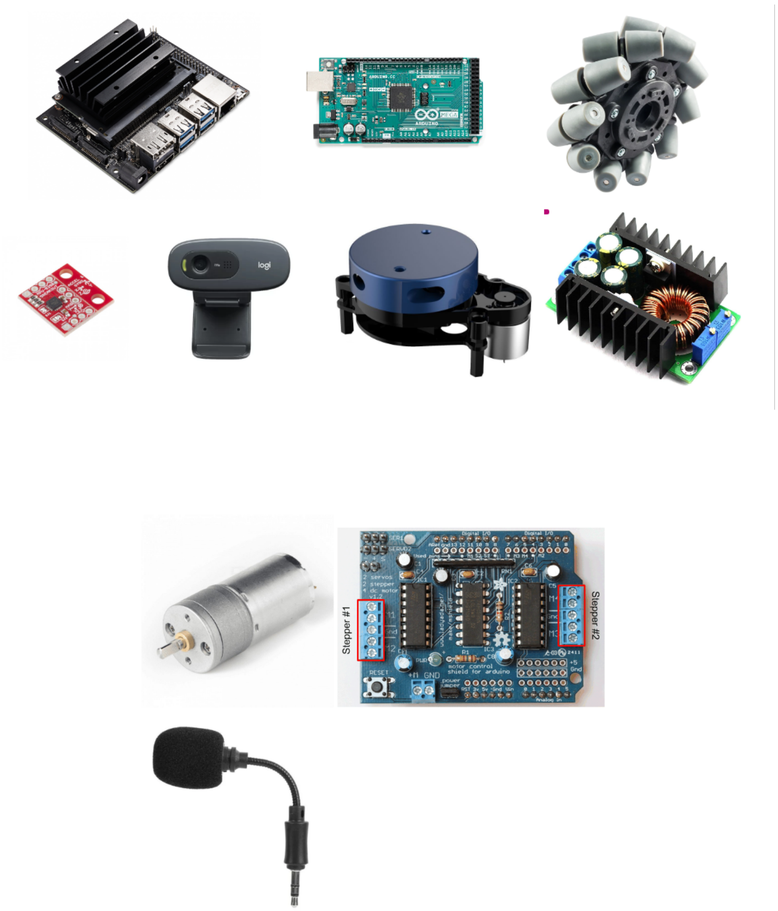


Figure 4.6: Necessary Equipments

4.3 Workplan

In order to explain the workflow of the Models used in our system a flowchart is presented. The purpose of the following flowchart in figure 4.7 is to demonstrate the flow of different functionalities that are working on our system. The functionalities that are mentioned are Recommendation, text-to-speech, and Summarizer. How the inner models of these functionalities are used in our chatbot has been visualized using the flowchart.

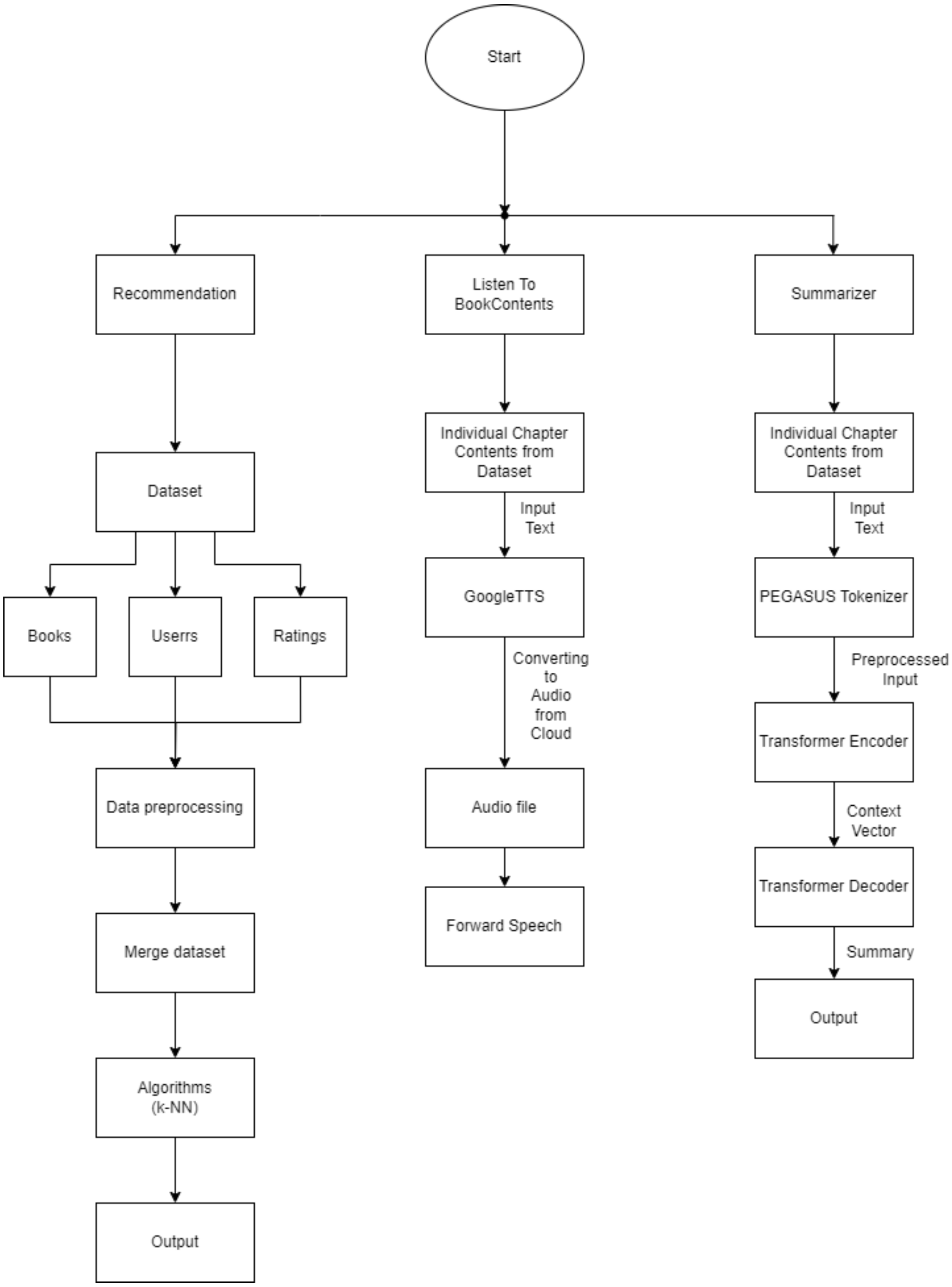


Figure 4.7: Model Workflow

In Figure 4.8, we have explained how our chatbot goes on to find answers to the user input. For different functionalities like recommendation, summarization, and reading out books the chatbot recognizes the input and then responds according to the user's desired input. If the chatbot is asked a question it would look for the answers in intents dataset or our scraped question answer dataset. If the inputs start to look like the chatbot responses are not satisfying for the user, with our sentiment analysis system we can bring in a human assistance.

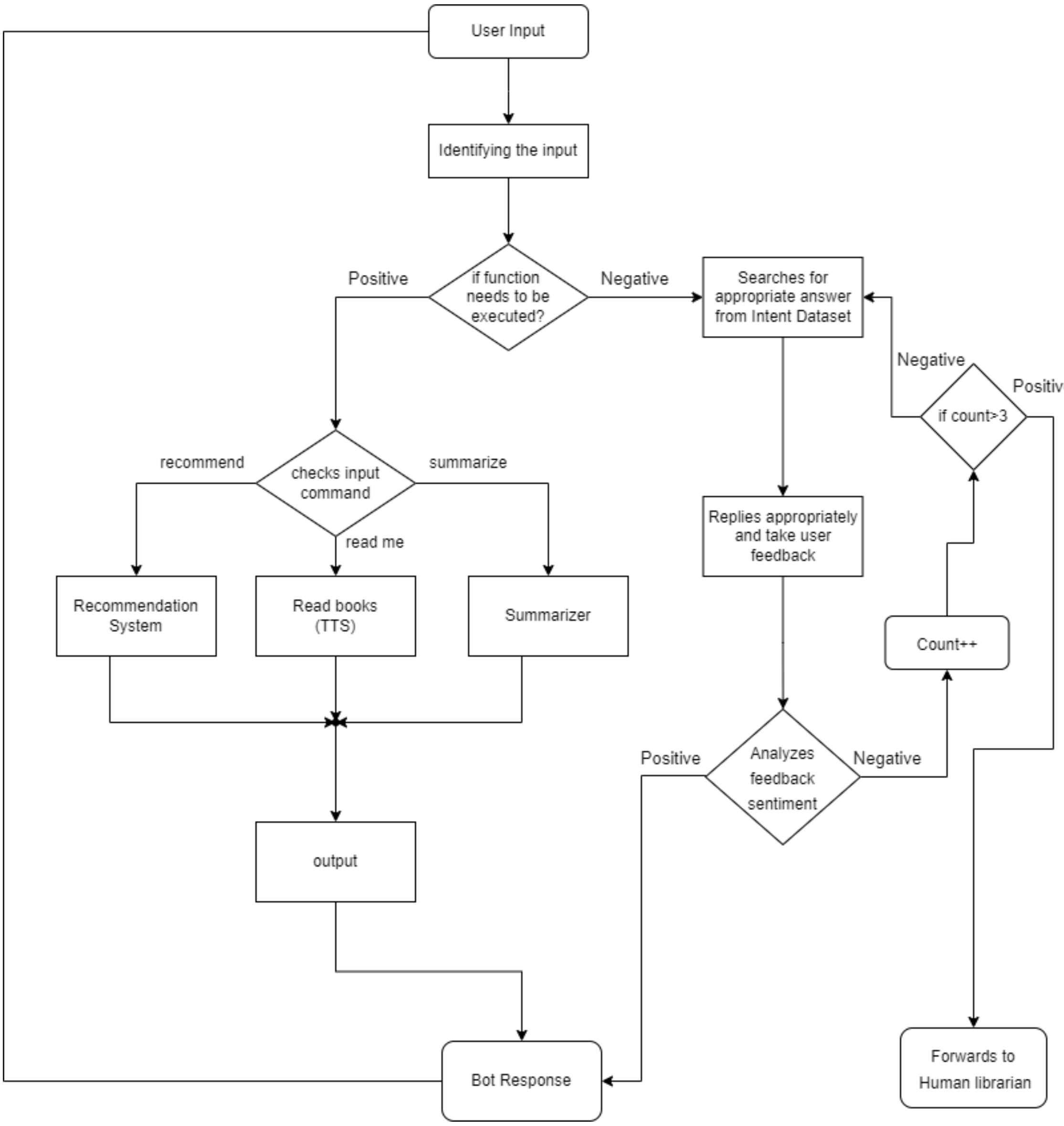


Figure 4.8: Chatbot Workflow

The robot's workflow visible in figure 4.9, consists of a number of interrelated procedures that are expertly coordinated to produce precise and efficient performance. The Jetson Nano is responsible for carrying out the first phase, which is the QR code detection. The Jetson Nano uses its computing capability to read and carefully scan QR codes that are in its range of vision. When a QR code is successfully scanned, PySerial communication is used to send the data contained therein to an Arduino Mega. Motor control is managed by the Arduino Mega, which is outfitted with an Adafruit motor shield. The robot can move and react in accordance with the instructions included in the QR code because the Arduino precisely moves the motors in response to the decoded data. The strong and competent automation system is ensured by the smooth integration of hardware components, which range from the high-performance Jetson Nano for visual tasks to the versatile Arduino Mega for motor control. This allows the robot to be responsive and adaptable in a variety of contexts.

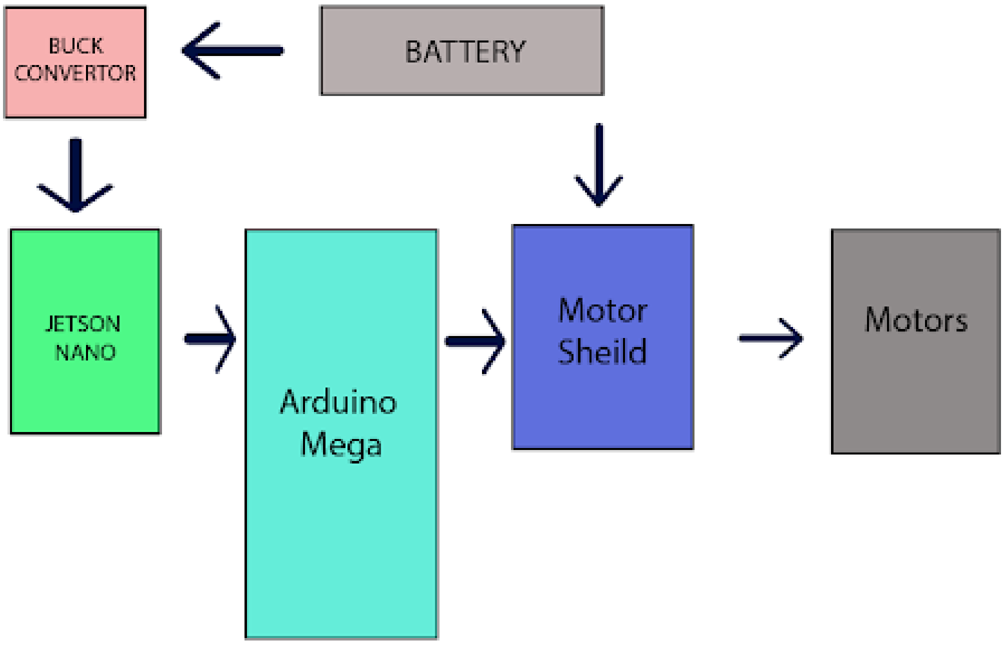


Figure 4.9: Work plan of the Robot

Chapter 5

Results, Analysis and Discussion

For a chatbot and a robot, the results depends on different metrics. In chatbots, the result depends on the performance and effectiveness of the chatbot to recognize intents, give better book recommendations, and give accurate summarization of different books and chapters. For a chatbot that is built to do multiple functionalities different matrices of result measurement are used.

5.1 Chatbot accuracy

For this analysis, we used the dataset that we specifically created for training our chatbot model. Table 3.1 provides examples of the many queries that are included in this dataset along with their corresponding routines. In the field of Natural Language Processing (NLP), transformer models have shown to be very useful. To train our dataset in this work, we used three well-known transformer models: BERT, RoBERTa, and DistilBERT. For each of the three models, the evaluation showed excellent performance, with measures like as accuracy, precision, and f1-score seemingly exceeding the 90% threshold. This demonstrates how well these transformer structures perform in handling our particular NLP problem.

5.1.1 Comparision between BERT, RoBERTa, and DistilBERT

Class	Precision	Recall	F1-Score	Support
others	0.75	1.00	0.86	6
tts	1.00	0.87	0.93	15
summary	1.00	1.00	1.00	9
recommendation	1.00	1.00	1.00	10
Accuracy			0.95	40
Macro Avg	0.94	0.97	0.95	40
Weighted Avg	0.96	0.95	0.95	40

Table 5.1: BERT Classification Report

After giving each model a thorough training, we find that every model performs above expectations. The performance metrics of BERT, RoBERTa, and DistilBERT

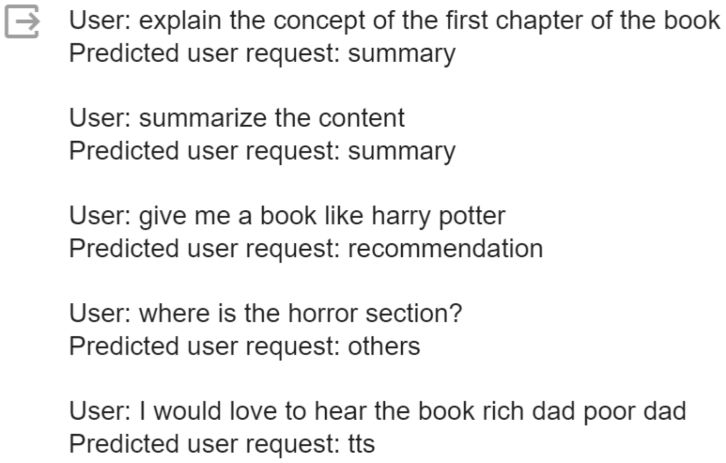
Class	Precision	Recall	F1-Score	Support
others	1.00	1.00	1.00	6
tts	0.94	1.00	0.97	15
summary	1.00	1.00	1.00	9
recommendation	1.00	0.90	0.95	10
Accuracy			0.97	40
Macro Avg	0.98	0.97	0.98	40
Weighted Avg	0.98	0.97	0.97	40

Table 5.2: RoBERTa Classification Report

Class	Precision	Recall	F1-Score	Support
others	0.75	1.00	0.86	6
tts	1.00	0.87	0.93	15
summary	1.00	1.00	1.00	9
recommendation	1.00	1.00	1.00	10
Accuracy			0.95	40
Macro Avg	0.94	0.97	0.95	40
Weighted Avg	0.96	0.95	0.95	40

Table 5.3: DistilBERT Classification Report

can be visually seen in Table 5.1, Table 5.2, and Table 5.3 respectively. However, upon closer examination from figure 5.2, we see that RoBERTa is more outstanding than the other two in terms of accuracy, precision, recall, and f1-score. Because of this, in the end, we were able to properly identify and evaluate user queries using RoBERTa, figuring out whether or not users are asking for the execution of particular functions. A sample of this prediction is shown on figure 5.1.



```
User: explain the concept of the first chapter of the book
Predicted user request: summary

User: summarize the content
Predicted user request: summary

User: give me a book like harry potter
Predicted user request: recommendation

User: where is the horror section?
Predicted user request: others

User: I would love to hear the book rich dad poor dad
Predicted user request: tts
```

Figure 5.1: Example of the prediction capability of user queries

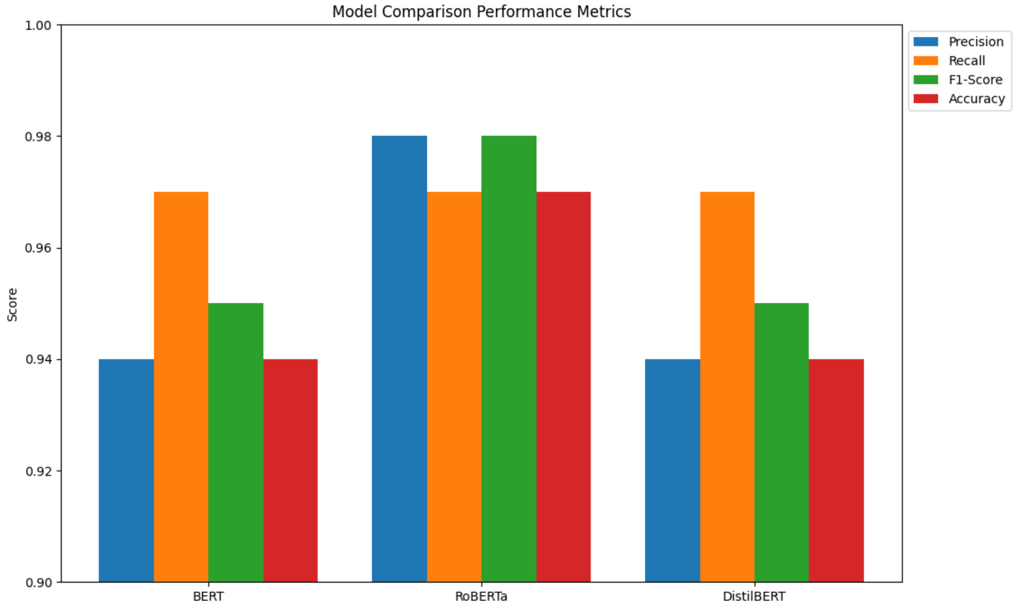


Figure 5.2: Comparision of the chatbot models

5.2 Summarizier accuracy

Titles, authors, and specific chapters of many books are all included in the JSON file that makes up the dataset for the books. We selected the book "Rich Dad Poor Dad" specifically for the purpose of our analysis. Three models—the BART model from Facebook, the PEGASUS model from Google, and the T5 (Text-to-Text Transfer Transformer) have been used for the summarizing task. Each model has used different configurations which are shown in table 4.1.

5.2.1 Evaluation Model

Two important measures are used to assess the summarization models' performance: the BLEU Score and the ROUGE Score. These metrics offer numerical assessments of the quality of the generated summaries in relation to a reference summary. Below is a brief explanation:

1. **Bilingual Evaluation Understudy (BLEU):** An indicator called BLEU is used to assess the quality of text produced by machines, especially in machine translation. Higher scores denote better similarity. Precision is calculated based on the existence of matching n-grams in the produced and reference texts.

2. **Recall-Oriented Understudy for Gisting Evaluation (ROUGE):** A set of measures called ROUGE is used to evaluate the quality of translations or text summaries. It uses metrics like ROUGE-N, ROUGE-L, and ROUGE-W to quantify the overlap of n-grams and word sequences between the produced and reference texts. Better similarity is indicated by higher ROUGE scores.

Model	BLEU Score	ROUGE-1	ROUGE-2	ROUGE-L
BART	0.005	0.0882	0	0.0882
PEGASUS	0.0132	0.1471	0.0233	0.1471
T5	0.0101	0.1176	0.0233	0.0882

Table 5.4: Evaluation Metrics for Different Models

5.2.2 Comparision between BART, PEGASUS, T5

Every algorithm looks for significant sentences, keywords, and words to use in order to produce a summary that conveys important information. Despite the fact that all three of them rely on transformers, their architectural designs differ. T5 presents a single text-to-text framework, BART has a bidirectional design, and PEGASUS mostly uses the decoder. T5 utilizes a text-to-text framework for different NLP tasks, BART uses a denoising autoencoder aim, and PEGASUS uses extracted gap sentences for pre-training. T5 is a text-to-text transfer framework that adds task prefixes to denote the kind of task. Task prefixes are not used in the same clear manner by PEGASUS and BART. After comparing the three summarization models, we have decided to utilize PEGASUS for chapter summaries. We came to this conclusion because, in both BLEU and ROGUE scores, PEGASUS was higher and better than the other two. A visual representation of the comparison can be seen in figure 5.3 and 5.4. Also, a numerical comparison can be seen in table 5.4. There is an example of the analysis of the summarizer function in figure 5.5, where we observe that it gives a summary of the whole chapter.

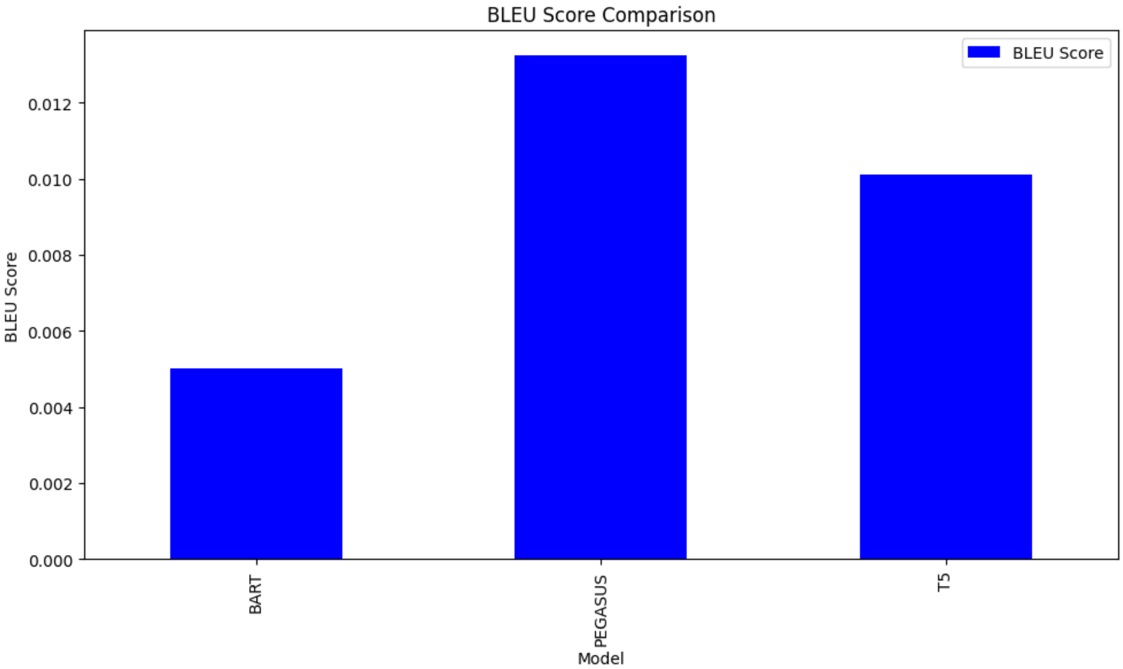


Figure 5.3: Comparision of BLEU Scores

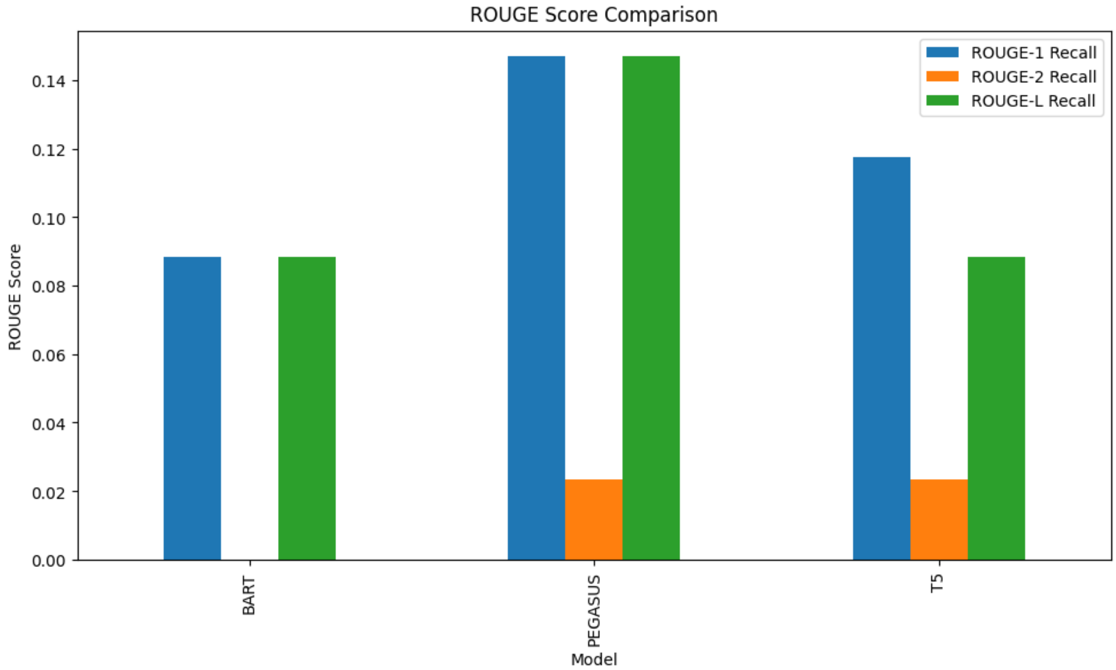
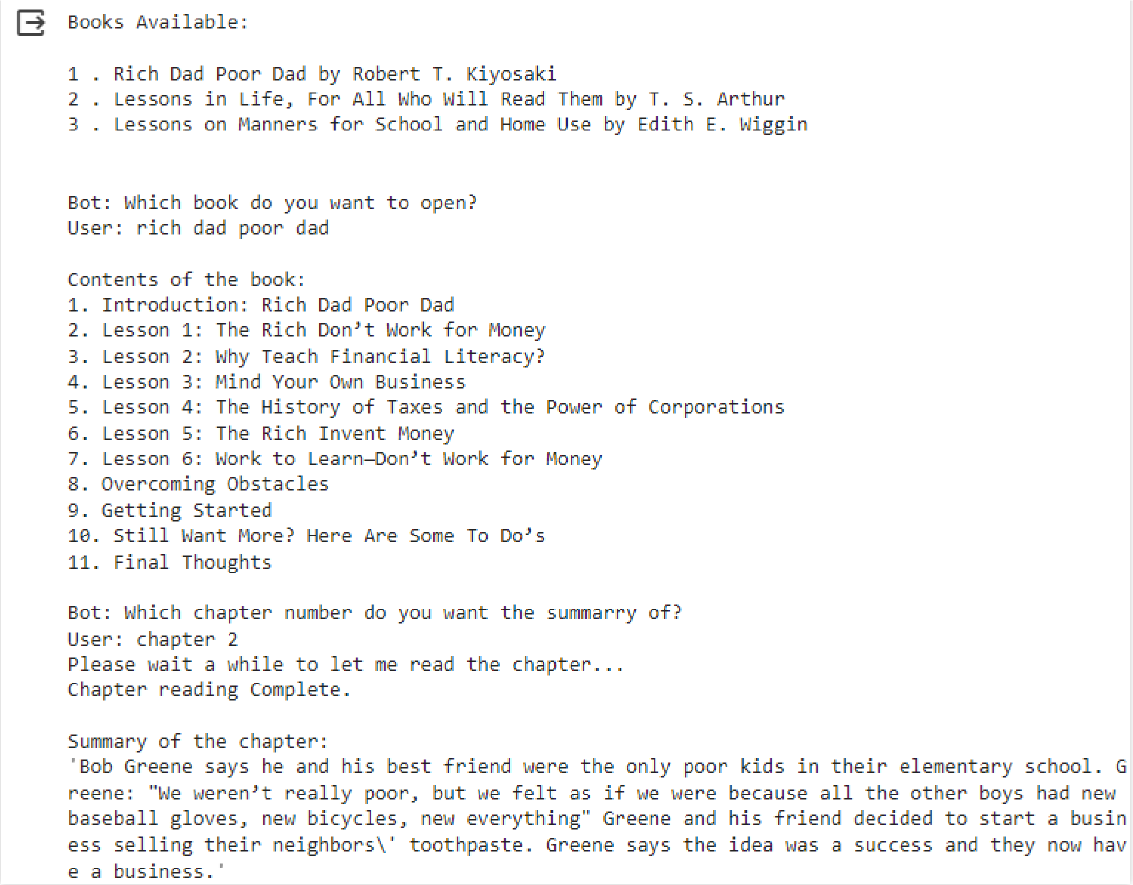


Figure 5.4: Comparision of ROGUE Scores



```
Books Available:

1 . Rich Dad Poor Dad by Robert T. Kiyosaki
2 . Lessons in Life, For All Who Will Read Them by T. S. Arthur
3 . Lessons on Manners for School and Home Use by Edith E. Wiggin


Bot: Which book do you want to open?
User: rich dad poor dad

Contents of the book:
1. Introduction: Rich Dad Poor Dad
2. Lesson 1: The Rich Don’t Work for Money
3. Lesson 2: Why Teach Financial Literacy?
4. Lesson 3: Mind Your Own Business
5. Lesson 4: The History of Taxes and the Power of Corporations
6. Lesson 5: The Rich Invent Money
7. Lesson 6: Work to Learn—Don’t Work for Money
8. Overcoming Obstacles
9. Getting Started
10. Still Want More? Here Are Some To Do’s
11. Final Thoughts

Bot: Which chapter number do you want the summarry of?
User: chapter 2
Please wait a while to let me read the chapter...
Chapter reading Complete.

Summary of the chapter:
'Bob Greene says he and his best friend were the only poor kids in their elementary school. G
reene: "We weren’t really poor, but we felt as if we were because all the other boys had new
baseball gloves, new bicycles, new everything" Greene and his friend decided to start a busin
ess selling their neighbors\' toothpaste. Greene says the idea was a success and they now hav
e a business.'
```

Figure 5.5: Example of Summarizer function

5.3 Recommendation accuracy

The basis of collaborative filtering algorithms is to find users with similar interests as the current user. The collaborative filtering algorithm does not give results based on the data itself, it gets the result from the data other users have provided. When a user wants a recommendation, collaborative filtering finds out what other books have good ratings according to other users that have shown interest in the book that is in input.

Evaluation model

For recommendation systems Mean deviation and Root mean squared errors are used to determine the accuracy of the system.
For this accuracy, a 20% ranking score is known as an acceptable rating, 21%-80% is not up to the mark and 81%-100% is known as a very low ranking [17].

Formula for Mean Deviation,

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n} |p_i - q_i| \tag{5.1}$$

In which, p_i is the predicted user rating, and q_i is the user's actual rating.

Formula for Root mean Squared error,

$$\text{RMSE} = \sqrt{\frac{1}{|T|}\sum_{(u,i)\epsilon T} (r_{ui} - \hat{r}_{ui})^2} \tag{5.2}$$

In which, T represents the test data, $|T|$ represents the size of the test data set, u represents the user, i represents the book, r_{ui} represents the user's actual score for the book, and $\hat{r}_{ui}$ represents the user's predicted score for the book.

Comparison between kNN and SVD

From the two models that we used for training, we have reached the conclusion that kNN has better recommendation accuracy than SVD. The difference between their recommendation accuracy is a lot in this case of our dataset. A visual representation between the comparison of the RMSE and MAE values of the models with the nearest neighbor values of 2, 5, 10, 20, and 50 can be seen in table 5.5. Also for a better understanding of the length of the difference we see the scatter plot diagram of RMSE and MAE in figure 5.6 and figure 5.7.

Model	**RMSE**	**MAE**
k-NN	0.9301174692989633	0.22584361523645333
SVD	3.3373727050707376	2.6601530920919267

Table 5.5: Results for book recommendation

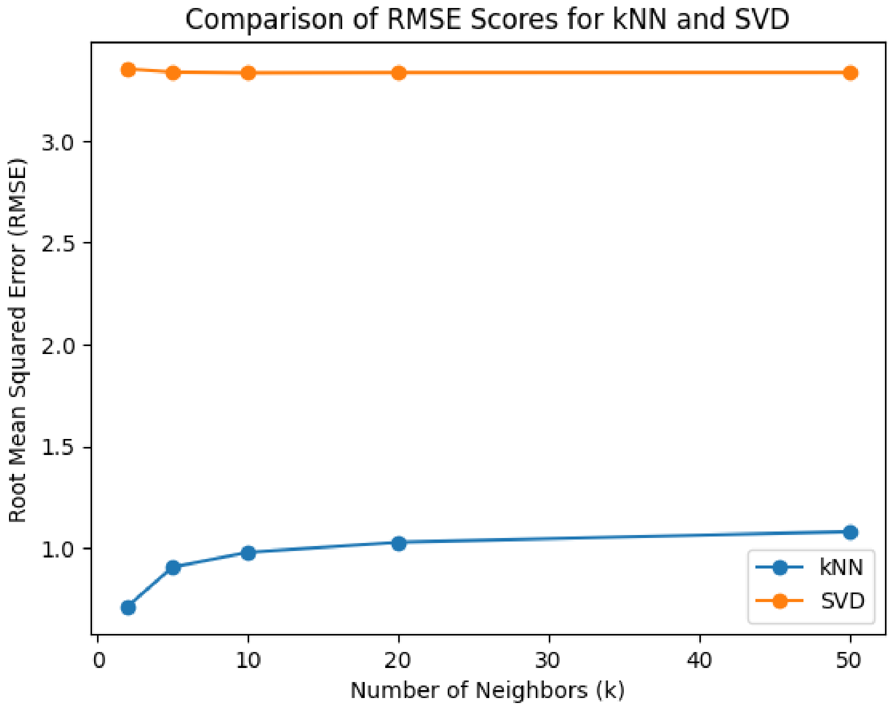


Figure 5.6: RMSE comparison between SVD and kNN

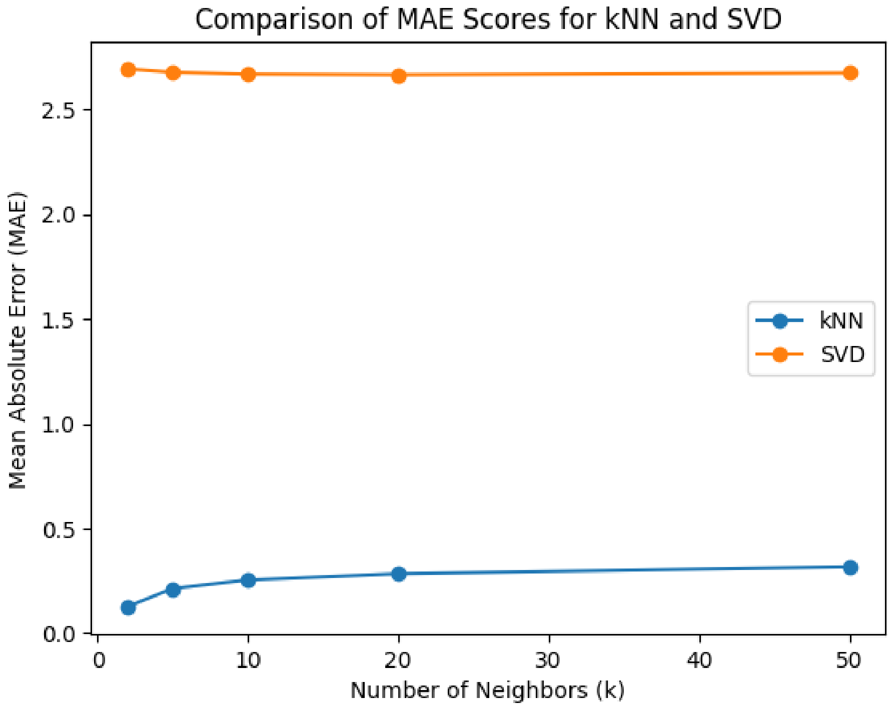


Figure 5.7: MAE comparison between SVD and kNN

5.4 Sentiment accuracy

Model Name	**Accuracy**	**Precision**	**Recall**	**F1 Score**
MultinomialNB	0.941	0.944	0.942	0.942

Table 5.6: Performance Metrics of Sentiment Analysis

The proposed model has demonstrated strong performance in the field of chatbots, surpassing the 90% threshold in terms of precision, accuracy, recall, and F1 scores. The effectiveness of the chatbot algorithm is demonstrated by the performance metrics that were acquired, which are displayed in tabular form in table 5.6 and as a bar chart in figure 5.8. Interestingly, the model consistently and accurately predicted answers to a wide variety of questions.

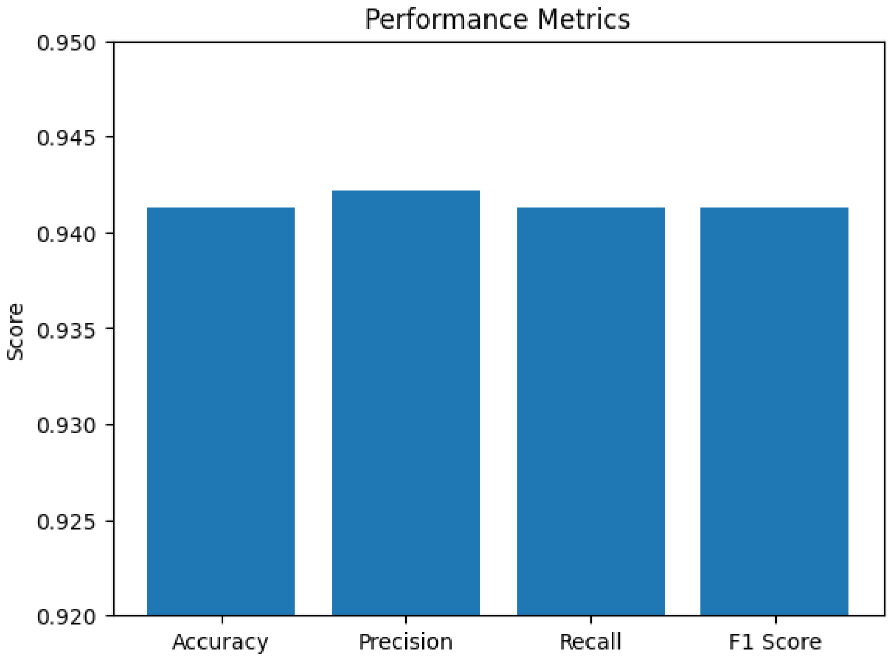
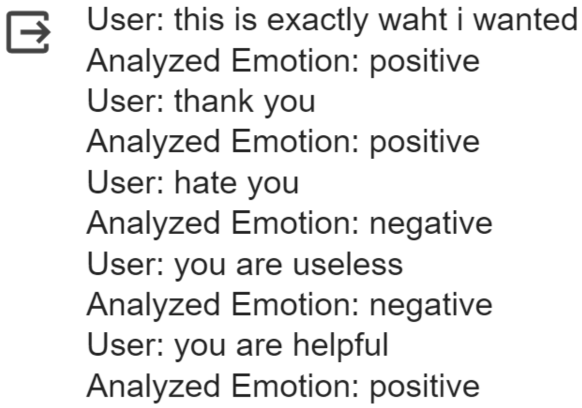


Figure 5.8: Performance Metrics of Sentiment Analysis

Figure 5.9 illustrates how effectively the system interprets the emotions expressed by users. This demonstrates the system's high degree of accuracy in identifying the emotions conveyed in user inputs. A thorough analysis of its prediction abilities will show how adept, robust, and adaptable the chatbot is at handling a variety of user inputs. The system's comprehension of every facet of the user's emotions is clearly illustrated by this graphic representation.



```
User: this is exactly waht i wanted
Analyzed Emotion: positive
User: thank you
Analyzed Emotion: positive
User: hate you
Analyzed Emotion: negative
User: you are useless
Analyzed Emotion: negative
User: you are helpful
Analyzed Emotion: positive
```

Figure 5.9: Predicting User Emotion

5.4.1 Analysis of Computer Vision System For Robot

The computer vision-based approach was conducted via two methods which have their unique characteristics and also unique how each functions.

1. Pattern recognition-based Approach: Firstly this approach exclaims the use of a pattern recognition algorithm provided by the Open Computer Vision (OpenCV) which identifies each unique QR code via its pattern recognition capabilities. The QRdetector algorithm is able to perform smoothly in regular conditions. Which may not be the condition in real-life scenarios. The main challenge the robot faces with this approach is that when the detection is conducted from a further distance the robot often does not recognize the QR code properly or provides false detections due to it being an algorithm-based approach and not having any proper context for the detections.

2. ML-based approach: We switched to an ML-based approach where a few popular pre-trained models were trained based on the custom dataset. The dataset consists of 10 classes each signifying different rows based on their QR code. The dataset consists of real-world sample images of the QR code collected from different distances to provide the ML algorithm with different scenarios. The data was then forwarded to the computer vision algorithm to properly classify each book without any sort of delay. With this approach, detections from various distances could be easily detected. With this approach setting the threshold allows the computer vision algorithm to adjust when required. Here we used the tensorflow framework and SSDMobile Net version 1. We got 86.16 accuracy.

Our custom-trained model takes the image feed via our generic USB camera. This then goes through the ML pipeline and afterwards, it classifies the image via the shelf number indicated by the image. As illustrated in the picture we can observe that the green bounding box defines the classification and the number in the center denotes the corresponding shelf number.
The robot first correctly identifies the Row number's QR, which in this case is 3, as seen in picture 5.10. The code is detected in the left portion of the screen by the

QR detection as well. Then, after identifying the QR code on the left in figure 5.10, it goes to the left appropriately and locates the code on the top screen shown in figure 5.11. After making some tweaks, the QR code was finally placed in the center of the camera. In picture 5.12, the robot is also pointing in the direction of the QR code to let the user know where they may find the book they're looking for.

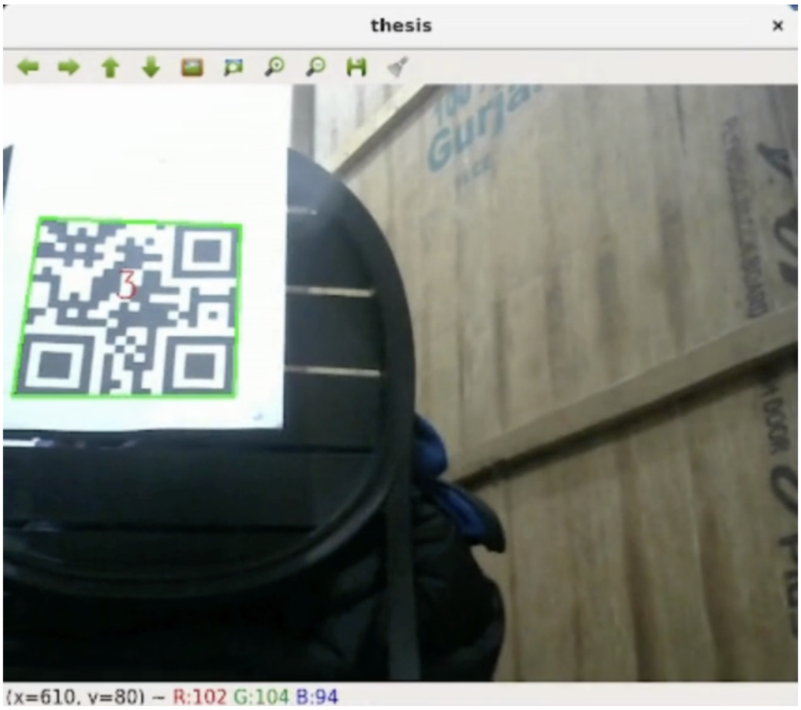


Figure 5.10: Detecting QR code coordinate at (99,100) - Signaling to go left.

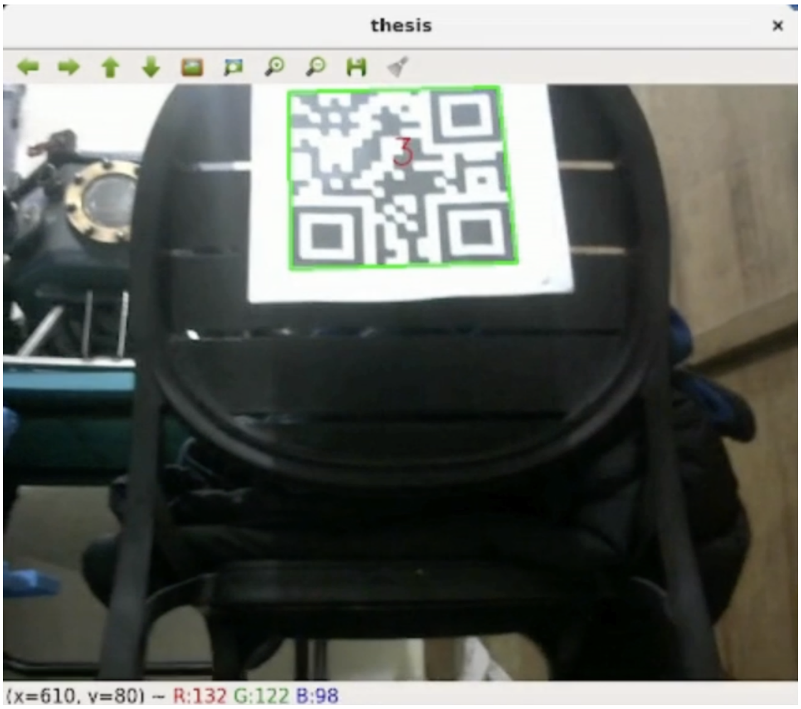


Figure 5.11: Detecting QR code coordinate at (320,74) - Signaling to go forward.

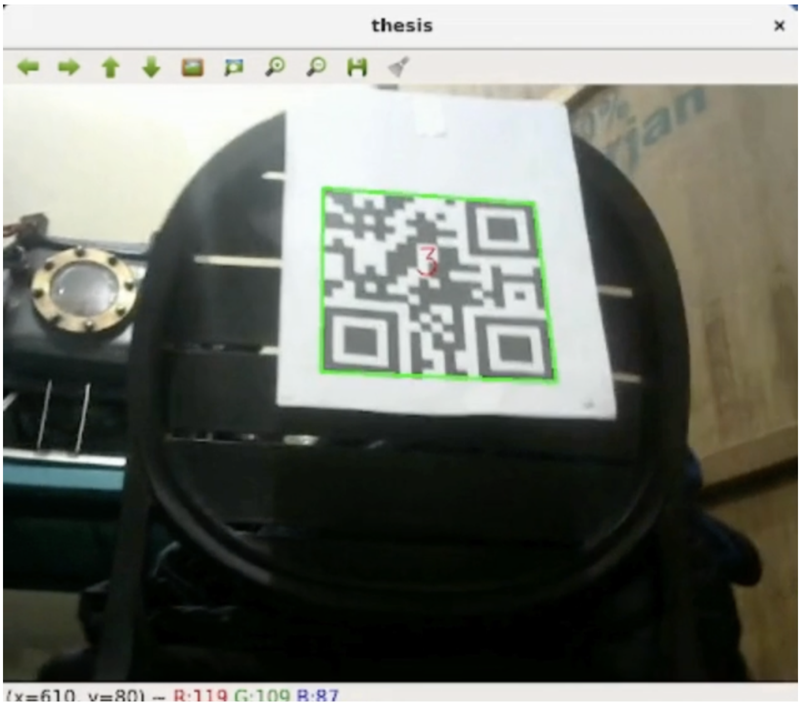


Figure 5.12: Detecting QR code coordinate at (337,160) - Signaling steady.

Chapter 6

Conclusion and Future Work

6.1 Conclusion

In the evolving world of information management and user engagement, the integration of a smart system into traditional library services emerges as a necessity, that has the ability of changing a conventional library into an intelligent one. With the help of an extensive dataset that we have gathered from several librarians as well as online for training, this system is expected to produce an outcome which will benefit everyone around it. To solve the complex library duties involved, we had divided the study into two essential parts including a hardware part and a software part. The software aspect has been implemented with an aim to improve the user experience. Conversely, the work towards the hardware component is aimed at optimizing the library experience, while ensuring a smooth and efficient interaction for library-goers to enjoy a seamless interface to navigate throughout the library. The integration of such a robot into the library infrastructure not only is able to enhance operational efficiency but also extend the library's accessibility. This is achievable due the possibility of continuous 24/7 operation with specialized assistance available at all times. And, this opens up the option for the library to be always open allowing enthusiast readers to come and go as per their need. The overarching goal here is to establish a library that transcends and overcomes the conventional boundaries. As technology advances, it was also appropriate to put it into practice for building a smart library. As a result, less time must be spent looking around a library, which improves efficiency and makes things run much more smoothly for both staff and visitors. This study believes a day where libraries effectively combine innovation with norms, maintaining their relevance and vitality in a constantly changing digital society.

6.2 Future Work

With the end of this research, there still exists the opportunity to further improve our workings to be able to fully implement our objectives. Building a proper overall assistant system that will be able to not only make the libraries smarter but also more interactive. Along with the collection and usage of diverse datasets which would help make the system more error-free and responsive. Also, by taking continuous user feedback and incorporating iterative improvements, the overall system can be further refined based on real world usage having diverse library environments. The system can also be further developed to support multiple languages, which will result in this to be implemented in many more libraries increasing the overall dataset and support for development. And with the collection of additional datasets, further analysis can be made to improve the overall efficiency of every library throughout. In essence, the future work outlined here envisions a continuous journey of improvement, adaptation, and collaboration.

Bibliography

[1] G. Guo, H. Wang, D. Bell, and Y. Bi, *KNN model-based approach in classification.* Aug. 2004.

[2] C. U. Press, *Stemming and lemmatization*, 2009. [Online]. Available: https://nlp.stanford.edu/IR-book/html/htmledition/stemming-and-lemmatization-1.html.

[3] D. G. Divayana, I. Putu, I. Made, and I. Wayan, “Digital library of expert system based at indonesia technology university,” *International Journal of Advanced Research in Artificial Intelligence*, vol. 4, Mar. 2015. DOI: https://doi.org/10.14569/IJARAI.2015.040301.

[4] A. Asemi, “Artificial intelligence (ai) application in library systems in iran: A taxonomy study,” Dec. 2016.

[5] A. Vaswani, N. Shazeer, N. Parmar, *et al.*, *Attention is all you need*, Dec. 2017. DOI: https://doi.org/10.48550/arXiv.1706.03762. [Online]. Available: https://arxiv.org/abs/1706.03762.

[6] M. Berbatova, “Overview on nlp techniques for content-based recommender systems for books,” Sep. 2019, pp. 55–61. DOI: https://doi.org/10.26615/issn.2603-2821.2019_009.

[7] K. Ganesan, *What are stop words?* Apr. 2019. [Online]. Available: https://www.opinosis-analytics.com/knowledge-base/stop-words-explained/#:~:text=Stop%20words%20are%20a%20set.

[8] A. Iwayemi and S. Adebayo, “Development of a robust library management system,” *International Journal of Computer Applications*, vol. 178, pp. 9–16, May 2019. DOI: https://doi.org/10.5120/ijca2019918850.

[9] M. Lewis, Y. Liu, N. Goyal, *et al.*, “Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension,” *arXiv (Cornell University)*, Oct. 2019. DOI: https://doi.org/10.48550/arxiv.1910.13461.

[10] S. Ray, *6 easy steps to learn naive bayes algorithm (with code in python)*, Sep. 2019. [Online]. Available: https://www.analyticsvidhya.com/blog/2017/09/naive-bayes-explained/.

[11] P. Vidhate, A. Badgujar, N. Patil, R. Pawar, and S. Mohammad, “A review on library automation using artificial intelligence,” *SSRN Electronic Journal*, vol. 6, pp. 301–303, Jan. 2019.

[12] K. Yu, R. Gong, L. Sun, and C. Jiang, “The application of artificial intelligence in smart library,” Jan. 2019. DOI: https://doi.org/10.2991/icoi-19.2019.124.

[13] S. S. MONDAL, *Collaborative book recommendation system*, Sep. 2020. [Online]. Available: https://www.kaggle.com/code/sankha1998/collaborative-book-recommendation-system/notebook.

[14] C. Raffel, N. Shazeer, A. Roberts, *et al.*, "Exploring the limits of transfer learning with a unified text-to-text transformer," *J. Mach. Learn. Res.*, vol. 21, no. 1, Jan. 2020, ISSN: 1532-4435.

[15] A. Menzli, *Tokenization in nlp: Types, challenges, examples, tools*, Jun. 2021. [Online]. Available: https://neptune.ai/blog/tokenization-in-nlp.

[16] A. Simha, *Understanding tf-idf for machine learning*, Oct. 2021. [Online]. Available: https://www.capitalone.com/tech/machine-learning/understanding-tf-idf/.

[17] Z. H. Wang and D. Z. Hou, "Research on book recommendation algorithm based on collaborative filtering and interest degree," *Wireless Communications and Mobile Computing*, vol. 2021, pp. 1–7, Jul. 2021. DOI: https://doi.org/10.1155/2021/7036357. [Online]. Available: https://www.hindawi.com/journals/wcmc/2021/7036357/.

[18] D. M. S. Barki, "Artificial intelligence applications and its impact on library management system by irjet journal - issuu," *International Research Journal of Engineering and Technology (IRJET)*, vol. 09, no. 09, Dec. 2022. [Online]. Available: https://issuu.com/irjet/docs/irjet-v9i9160.

[19] *Project gutenberg*, https://www.gutenberg.org/, Accessed: December 28, 2023, 2023.

[20] M. Wang, P. Xie, Y. Du, and X. Hu, "T5-based model for abstractive summarization: A semi-supervised learning approach with consistency loss functions," *Applied Sciences*, vol. 13, no. 12, p. 7111, Jan. 2023. DOI: https://doi.org/10.3390/app13127111. [Online]. Available: https://www.mdpi.com/2076-3417/13/12/7111#:~:text=The%20transfer%20text%2Dto%2Dtext.

[21] A. W. IBRAHIM, *Emotion dataset*. [Online]. Available: https://www.kaggle.com/datasets/abdallahwagih/emotion-dataset.

[22] Z. Kuang, *Singular-Value Decomposition and its Applications*. [Online]. Available: https://math.ucsd.edu/sites/math.ucsd.edu/files/undergrad/honors-program/honors-theses/2017-2018/Zecheng_Kuang_Honors_Thesis.pdf.

[23] *Pegasus: Pre-training with extracted gap-sentences for abstractive summarization*. [Online]. Available: http://arxiv.org/abs/1912.08777.