Project Report on

Synapse: Where AI Meets Education

Submitted in partial fulfillment of the requirements for the award of the degree of

Bachelor of Technology

in

Computer Science and Engineering

By

Fathima Jennath N K (U2103089)
Gautham C Sudheer (U2103092)
Godwin Gino (U2103096)
Mohammed Basil (U2103139)

Under the guidance of

Dr. Mary Priya Sebastian

Department of Computer Science and Engineering
Rajagiri School of Engineering & Technology (Autonomous)
(Parent University: APJ Abdul Kalam Technological University)
Rajagiri Valley, Kakkanad, Kochi, 682039
April 2025

CERTIFICATE

This is to certify that the project report entitled ***Synapse: Where AI Meets Education*** *is a bonafide record of the work done by* ***Fathima Jennath N K (U2103089)****,* ***Gautham C sudheer (U2103092)****,* ***Godwin Gino (U2103096)****,* ***Mohammed Basil (U2103139)****, submitted to the Rajagiri School of Engineering & Technology (RSET) (Autonomous) in partial fulfillment of the requirements for the award of the degree of Bachelor of Technology (B. Tech.) in "Computer Science and Engineering" during the academic year 2024-2025.*

Dr. Mary Priya Sebastian
Project Guide
Associate Professor
Dept. of CSE
RSET

Ms. Anu Maria Joykutty
Project Coordinator
Assistant Professor
Dept. of CSE
RSET

Dr. Preetha K G
Head of the Department
Professor
Dept. of CSE
RSET

ACKNOWLEDGMENT

We wish to express our sincere gratitude towards **Rev. Dr. Jaison Paul Mulerikkal CMI**, Principal of RSET, and **Dr. Preetha K G**, Head of the Department of Computer Science and Engineering for providing us with the opportunity to undertake our project, "Synapse: Where AI Meets Education".

We are highly indebted to our project coordinators, **Ms. Anu Maria Joykutty**, Assistant Professor, Department of Computer Science and Engineering, **Dr. Jisha G**, Associate Professor, Department of Computer Science and Engineering, for their valuable support.

It is indeed our pleasure and a moment of satisfaction for us to express our sincere gratitude to our project guide **Dr. Mary Priya Sebastian** for her patience and all the priceless advice and wisdom she has shared with us.

Last but not the least, we would like to express our sincere gratitude towards all other teachers and friends for their continuous support and constructive ideas.

Fathima Jennath N K
Gautham C Sudheer
Godwin Gino
Mohammed Basil

Abstract

This project seeks to revolutionize the educational experience with an innovative AI-powered web platform designed to convert text images—such as pages from textbooks—into editable and searchable digital content. By leveraging cutting-edge Optical Character Recognition (OCR) and Natural Language Processing (NLP) technologies, this platform aims to enhance note-taking and learning. The integration of these advanced technologies promises to make educational materials more accessible and interactive, providing users with a powerful tool to streamline their study and teaching practices.

One of the core features of this project is its OCR-powered text extraction capability, which allows users to convert various text images into editable digital notes. This feature supports a range of fonts, languages, making it highly versatile. In addition to text extraction, the platform includes an AI-powered chatbot that enables users to interact with their notes in a conversational manner. This chatbot can answer questions, generate summaries, and assist with note organization, further enhancing the learning experience. Another significant aspect of this project is its multilingual support. The platform's translation capabilities cater to international students and diverse linguistic backgrounds, ensuring that educational content is accessible to users regardless of their primary language. Additionally, features like text-to-speech functionalities support interactive learning and accessibility, making the platform adaptable to various learning needs.

By addressing the critical need for efficient and accessible educational tools, Synapse aims to bridge educational gaps and promote inclusive learning. Its user-friendly, scalable, and secure design makes it a valuable resource for students, educators, and institutions worldwide. The platform is set to foster a more effective and engaging learning environment, leveraging AI technologies to create a more dynamic and inclusive educational experience.

Contents

List of Abbreviations

OCR-Optical Character Recognition

CNN-Convolution Neural Network

FCN-Fully Convolutional Network

BP-Back Propagation

LLM-Large Language Model

RAG-Retrieval Augmented Generation

TAM-Technology Acceptance Model

LSTM-Long Short-Term Memory

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background

This project is an innovation in the AI-driven platform aimed at changing the education experience through digital capabilities. The project addresses problems that students and educators face when converting physical, multilingual content into editable and searchable formats. Conventional forms of digitizing educational material can be very cumbersome and take much time to execute. Synapse makes this process easier and quicker through integration with sophisticated technologies such as Optical Character Recognition (OCR) and Natural Language Processing (NLP).

Synapse's OCR feature allows a user to convert text images—such as those from textbooks—into editable digital formats. This functionality supports various fonts and languages, making it flexible and versatile. Additionally, it includes an AI-powered chatbot that allows users to engage in conversation and perform functions like question answering, note organization, and summarization. These features make the learning and studying process more efficient and effective in education.

Another feature is that it supports several languages, allowing the content to be accessible for users from other linguistic backgrounds. The platform includes text-to-speech feature, which also improve accessibility and adaptability according to different needs of learning; all users are benefited, whether they have different learning preferences or abilities.

In a world where digital, interactive learning tools are in great demand, Synapse stands out as a solution that promotes inclusive and efficient education. Its ability to bridge the gaps in educational disparities, facilitate inclusive learning, and provide an interactive environment makes it valuable for students, educators, and institutions alike. This creates a user-friendly, interactive environment. Synapse is a safe, scalable solution that leverages AI to build a dynamic, captivating learning environment that meets the demands of contemporary education.

1.2 Problem Definition

This is a project that aims at building an AI-based platform to solve the problem of different educational content, such as text, images, or multilingual materials, being converted into a structured, interactive, and flexible digital form. With features like AI-driven chatbots, multilingual support, and text-to-speech functionalities, the platform meets the needs of students and educators, promoting inclusive education.

1.3 Scope and Motivation

Scope: The scope of Synapse includes designing an AI system that accurately recognizes, processes, and converts text from images and multilingual content into editable digital content. Furthermore, Synapse will provide virtual assistant capabilities, including chatbots for interaction, translation, and text-to-speech for students with learning difficulties. The platform should be applicable across various educational institutions to allow students to learn both individually and in groups. In the long run, Synapse plans to improve the learning process by making educational material more structured and accessible.

Motivation: The rationale for Synapse lies in the need for increased availability of digital and accessible educational content in a society that has become rather diverse. The conventional approaches to learning entail lengthy courses of writing notes and sorting content, which prove impractical for students and instructors. Secondly, there is an urgent demand for learning technologies that accommodate multilingual and multimodal learning needs in the contemporary interconnected world based on technology. By leveraging AI, Synapse seeks to address accessibility and engagement challenges, narrowing learning gaps while accommodating diverse educational requirements. This project aims to provide easy access to learning materials worldwide and promote more interconnected educational content.

1.4 Objectives

1. Create a platform that uses OCR technology to transform text, photos, and multilingual instructional materials into digital formats that can be edited and searched.
2. Bring in AI-powered chatbots that allow users to ask questions and organize notes and summaries, among other things.
3. Supporting users of different linguistic backgrounds by which to help create an inclusive education.
4. Provide text-to-speech feature to ensure ease of access for people with special needs.
5. We develop a scalable, handheld, user-friendly interface that can be used by educational institutions as well as individual users, making it easy as well as extensible.

1.5 Challenges

Accurate extraction of text from various input types and comprehending technical terminologies can generate erroneous outputs. High accuracy for translations across multiple languages poses a significant difficulty, especially for diverse linguistic systems. Handling heavy volumes of data and concurrent requests without degrading performance also requires a scalable architecture. The additional AI-based functionalities can also pose the problem of consuming large computing resources, with potential high costs of operations.

1.6 Assumptions

The users are required to input clear, high-resolution images so that the text is accurately read by the OCR module. Strong and efficient language models, as well as translation algorithms, must also be required by it. Sufficiently consistent and high-speed internet is considered for carrying on smooth communication of the backend modules with AI. Regular updates and model training are also assumed in the AI-powered chatbot and NLP system.

1.7 Societal / Industrial Relevance

Synapse can have a big impact on our society and economy, especially in areas like education, making content digital, and improving access to information. In schools and colleges, Synapse can change the way students and teachers use learning materials. It can provide content in different languages and cultures, so more people can understand and use it. Students can also learn more effectively with features like chatbots, question-answer generation, and summarization tools, which make studying easier and more interactive.

At an industrial level, the platform can be employed by organizations to digitize their documents for more efficient database management, thereby reducing operating time and workload. Language translation between spoken and written forms—as well as between text and speech—is a core feature of Synapse, which makes learning more accessible for individuals with diverse learning capabilities. Lastly, Synapse helps to improve productivity, increase access to information, and contribute meaningfully to the digitalization process.

1.8 Organization of the Report

- **Chapter 1:** This chapter covers the project's background, problem definition, scope, objectives, relevance, assumptions, and challenges.
- **Chapter 2:** This chapter surveys literature on methodologies in text recognition, machine translation, educational chatbots, and text summarization, assessing their effectiveness, limitations, and applications to highlight potential improvements in text processing technologies.
- **Chapter 3:** This chapter covers the system architecture, module division, expected output, software and hardware requirements, and project timeline for the Synapse platform, providing a comprehensive overview of its design and implementation.
- **Chapter 4:** This chapter includes the implementation of the system.
- **Chapter 5:** This chapter includes the results and discussion of the project.
- **Chapter 6:** This chapter comprises the conclusion and the future scope of the project.

1.9 Conclusion

This chapter introduced the project by describing its background, defining the main problem, and describing its scope and motivation. It stated the project's objectives, emphasized the challenges, and listed key assumptions. It also explained why the project matters to both industry and society, setting the stage for the rest of the work.

Chapter 2

Literature Survey

2.1 Two-Stage CNN Approach for Text Line Recognition in Camera-Captured Images

2.1.1 Introduction

This paper proposes a two-step CNN that is dedicated to text line recognition on mobile and embedded devices. The method is light-weight and very effective for low complexity applications, and does not require any knowledge of language models. The current model succeeds in segmenting text lines into character level before recognition, which makes it more language and text style insensitive. Examples of use cases are, smart glasses, mobile translators and other devices that need on-substrate text analysis. This approach strives to achieve moderate computational reliability on reasonable accuracy. [1]

2.1.2 Methodologies

The model utilizes a two-step CNN process: the first type cuts text lines into individual characters and the second type recognizes these characters. Segmentation is treated with dynamic programming to reduce usage of language models and improve efficiency. The two-step design allows segmentation and recognition to be performed in real time when using mobile devices. Dynamic programming takes it a notch higher and optimizes segmentation is most accurate in text that has a rather simple layout, so additional mistakes appear to be untypical for this tool. This setup makes the framework Light allowing for its use in embedded systems by not wasting time and resources. The below figure describes the text line recognition scheme:

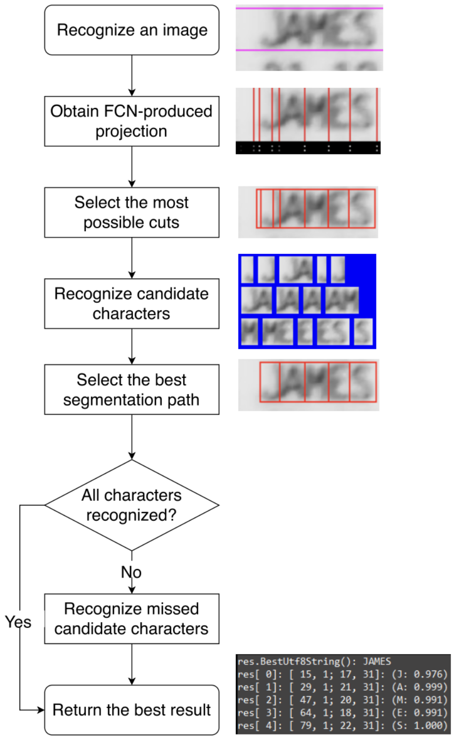


Figure 2.1: Text line recognition scheme.

2.1.3 Advantages

- Lightweight and efficient, ideal for low-resource mobile and embedded devices.
- Language-independent, enabling usage across various languages and text structures.
- Real-time segmentation without relying on language-specific models, boosting adaptability.
- High speed, balancing computational load with accurate character recognition.
- Suitable for mobile text recognition applications requiring on device processing.

2.1.4 Disadvantages

- Lack of post-processing can affect accuracy in complex text scenes.
- Struggles with cursive or connected characters due to segmentation focus.
- Accuracy decreases in scenes with overlapping or distorted characters.
- May be less effective in noisy environments where background interference is high.

- Limited flexibility in complex text layouts due to reliance on simple segmentation.

2.1.5 Results

The two-step CNN framework showed a 30% improvement in speed over traditional models, making it highly efficient for embedded devices. Segmentation accuracy was high for simple text cases but dropped by 10-15% in complex scenes, particularly those with connected or cursive characters. Its lightweight design facilitated real-time processing, supporting on-device applications with minimal delay. The model's adaptability to multiple languages without language-specific models contributed to its success in diverse text cases. While effective, additional post-processing techniques could improve accuracy in challenging text environments. The paper concludes that the two-step CNN framework is highly effective for resource constrained applications, achieving a favorable balance between accuracy and computational efficiency. While it excels in segmenting and recognizing text on mobile devices, its lack of post-processing limits performance in more complex text scenarios. Future work could focus on refining post-processing methods to improve accuracy in challenging environments. The language-independent nature of the model holds promise for multilingual applications. Overall, the approach demonstrates significant potential for embedded device applications, though improvements are needed to boost accuracy in difficult cases.[2][3][4]

2.2 Neural Network-Based English Machine Translation System

Considering the current issues of low efficiency of machine translation and quality of translations, the development of this paper employs a research method. Concerning evaluation of English translation machine employing a Back Propagation (BP) neural network algorithm. In this case, the method enhances the knowledge of users, making the machine translation service to be more smarter. By applying the function of BP neural network algorithm, for the English online translation as research object, Google translator has the highest quality, they only have error frequency 167, while the translation error rate of Baidu and iFLYTEK is 266 and 301 respectively, which is significantly higher than Google translation. To serve this requirement, a model of machine translation evaluation system based on the explicit neural network algorithm is presented. It overcomes the problems of the conventional English machine translation. The result indicates that the machine translation system developed under the framework of the neural-network algorithm can improve on the inherent weakness of machine translation, including the improper usage of information and large scale of model parameters, and further improve the performance of the neural network machine translation.

The machine translation method described in this paper leverages neural networks, particularly the Back Propagation (BP) neural network algorithm, to evaluate and improve translation accuracy. This approach is built on deep learning concepts, specifically language models that capture the relationships between words to generate fluent, contextually relevant translations. This report describes each methodological step involved, explaining both the theoretical rationale and the mathematics behind each process.[5]

2.2.1 Methodology

Word Representation

The first step in developing a neural network-based translation system is representing words numerically in a way that preserves their meaning and relationships with other words. Traditional machine translation models often rely on *One-Hot Encoding* for word representation, which uses high-dimensional vectors with mostly zeros and a single one to denote the word's identity. However, *Word Embeddings* (or word vectors) are used in neural networks to represent each word as a low-dimensional, dense vector that encodes its meaning and relationship to other words.

In One-Hot Encoding, each word w in a vocabulary of V words is represented as a vector of length V, with all values zero except one. However, such vectors are sparse and do not capture semantic similarity. Instead, in word embeddings, similar words have vectors close to each other, which is often calculated using *Euclidean distance* or *cosine similarity.* This step is crucial because it translates discrete language data into a format that the neural network can process, enabling it to recognize word similarities and differences.

Probabilistic Language Modeling

A language model assigns probabilities to sequences of words, which helps the model understand likely word orders and relationships in a sentence. This model forms the foundation for generating grammatically and contextually accurate translations.

For a sequence of words $w = (w_1, w_2, \ldots, w_t)$, we define the *probability of the sequence* as:

$$P(w) = P(w_1)P(w_2|w_1)P(w_3|w_1, w_2)\ldots P(w_t|w_1, \ldots, w_{t-1}) \tag{2.1}$$

This equation implies that the probability of a sequence can be decomposed into the product of conditional probabilities of each word given all preceding words. Here, each term $P(w_i|w_1, \ldots, w_{i-1})$ represents the probability of word w_i occurring given the previous words in the sequence, enabling the model to predict the next word based on context.

In practice, to simplify calculations, we apply the *Markov assumption*, which posits that the probability of a word depends only on a fixed number of preceding words, not the entire sequence. This reduces the model to:

$$P(w) = P(w_1)P(w_2|w_1)P(w_3|w_2)\ldots P(w_t|w_{t-1}) \tag{2.2}$$

This version considers only the previous word for each term, making computations feasible. The assumption is useful in capturing local dependencies and allows the model to learn transitions between adjacent words, albeit at the cost of losing long-term dependencies.

Neural Probabilistic Language Model

The neural probabilistic language model overcomes the limitations of the n-gram model by using a neural network to directly estimate the probability distribution of a word given its context. This model leverages neural networks to learn and generalize patterns across sequences.

For a given word sequence, we define a *conditional probability*:

$$P(w_t|w_{t-n+1}, \ldots, w_{t-1}) = f(w_t, w_{t-1}, \ldots, w_{t-n+1}) \tag{2.3}$$

where f is a function approximated by a neural network that learns the joint probability distribution of words in the language. In this case, based on the training process, other hidden parameters such as those related to weight in the hidden layers of the network is fine tuned so as to achieve optimal output of the model in terms of word sequence prediction.

Transformer Model and Self-Attention Mechanism

The Transformer model incorporates a *self-attention mechanism* for capturing the dependency across a sequence to an extent that it is not bound by sequence length. In MT this is important for producing good translations since in most cases the meaning of a word is closely associated with other words that may be far from it in the sentence.

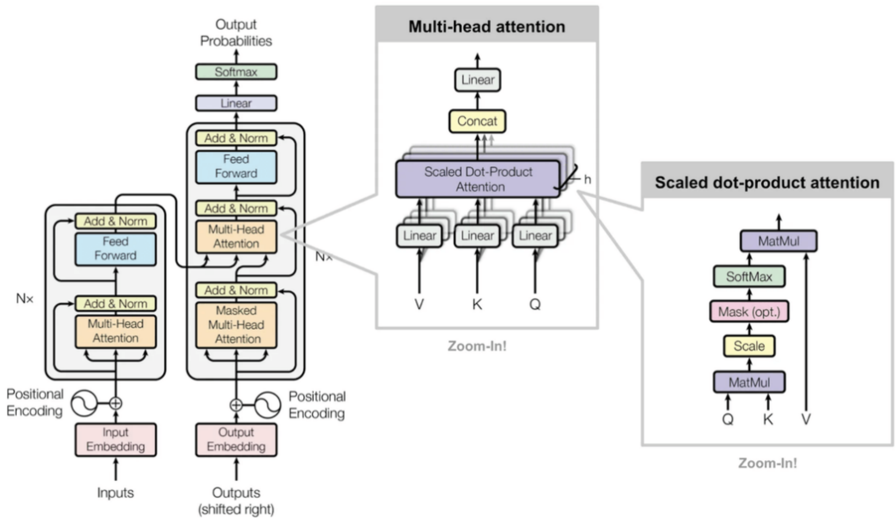


Figure 2.2: Transformer Model and Self-Attention Mechanism Framework

In the Transformer, the source language $X = (w_1, w_2, \ldots, w_m)$ and target sequence $Y = (y_1, y_2, \ldots, y_m)$ that undergo self-attention and feed-forward networks of multiple layers. Each layer produces a weight for each word in the sequence according to its correlation to all the other words, as is important for correct translation. The write attention score between words is computed using query (Q), key (K) and value (V) vectors:

$$Attention(Q, K, V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right) \tag{2.4}$$

Here, dk mentioned is the scaling factor that is generally, dimensionality of the key vectors is used to normalize the dot product. This equation sums up an array of values multiplied by a corresponding array of weights that at this point reflect the degree of similarity between query and key vectors. It is sometimes used to normalize the weights where they sum up to one; we can interpret the weights as probabilities.

This Transformers model allows the neural network to selectively inspect the passages for words that provide important context to it and thus very enhance the translation accuracy.

Training with BP Neural Network

BP is used for training the neural network since the aim is to minimize the prediction error, it is done iteratively. This weights adjustment occurs, during the training phase, by determining the gradients of the error with regard to the weights and moving the weights using gradient descent. In the case of a training set, the BP algorithm calculates the error between the output calculated by the network and the output of the training data and then passes this error back through the network to reduce it for future instrumental results. This process is performed several epochs in order to find an optimal weights that minimize this loss function.The idea is to eventually discover the best parameters against the error when trying to predict a series of words. This training process helps the model to generalize correctly the interaction between the word pairs in the source and target languages.[6]

2.2.2 Final Insights and Implications

The present work introduces an evaluation system of English machine translation which employs a BP neural network that aims at translation efficiency and effectiveness. Most rule-based as well as statistical translation methods fail on the issue of heavy feature engineering by humans, with their inability to make accurate decisions about difficult language structures. The paper uses a model for the first time based on the neural network model, so the translation becomes very fluent and has less inaccuracies. Based on this model, the BP neural network analyzes and adjusts translations through feedback of error. It's trained on a corpus that encompasses many translations through the platforms such as Google, Baidu, and iFLYTEK in comparing machine output to human reference translation. In doing so, the errors were divided into three ontological, semantical, and syntactic groups, enabling error patterns and weakness to be identified systematically. This type of iterative learning process improves model performance and translates into the sophisticated assessment of both translation accuracy and fluency. Translation accuracy and fluency of traditional models can also be compared when using a more advanced system for translation that incorporates neural networks: Superior translation accuracy on complex language tasks, long sentences, and culturally charged phrases.[6]

2.3 LLM-Powered Chatbot for Databases and Information Systems in Higher Education

This paper presents the design and implementation of MoodleBot, a big language model (LLM)-powered chatbot, for use in higher education to facilitate learning on databases and information systems. The MoodleBot integrates into Moodle, an LMS commonly used in learning environments, to provide self-regulated learning (SRL) and help-seeking functionalities. MoodleBot is designed to provide students with live support and boost engagement while it reduces some bureaucratic burdens for teaching staff. A specific aspect analyzed in this article is how well students find a chatbot as useful and, at the same time, estimates its correctness while matching course materials.[7]

2.3.1 Methodologies

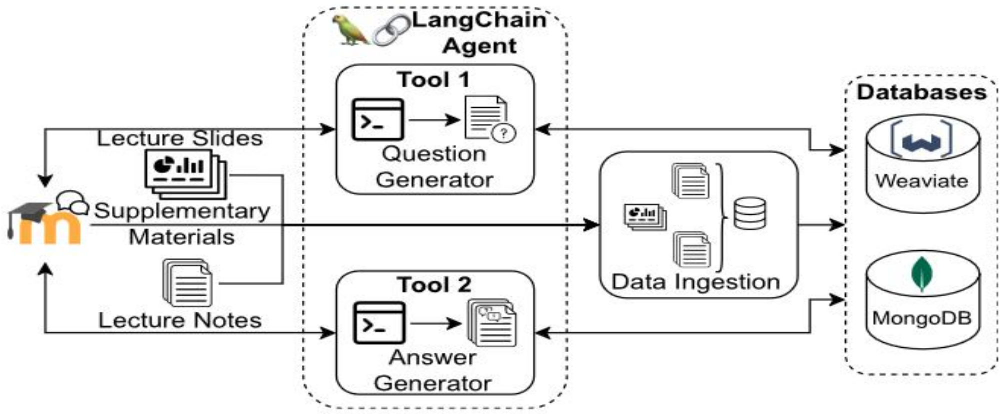


Figure 2.3: MoodleBot's architecture

Fig 2.3 below represents the architecture of the MoodleBot.

1. Data Acquisition and Vectorization:

- Source of Content:MoodleBot is a bot that pulls content for a specific course, from things like lecture notes, PDFs, slides, etc. The information that MoodleBot draws upon in providing accurate responses to student questions are included in these materials.
- Text Chunking and Embedding: First, the collected content is processed into (manageable) text chunks. With this segmentation the chatbot is able to focus on a few parts of the text instead of the entire text, and thus will be able to provide faster, and more contextually relevant responses.
- Embedding with OpenAI's Model: Embed each text chunk into high dimensional vectors with OpenAI's text embedding ada 002 model. By these embeddings, we can use the content as a semantic searching format.
- Storage in Weaviate Vector Database:Then, once vectorized, these embeddings are stored in a Weaviate vector database. The purpose of this database is to facilitate a fast semantic search and make it possible for MoodleBot to rapidly gather and utilize background knowledge.

2. Integration with Moodle LMS:

   MoodleBot is directly embedded within Moodle's chat interface, providing a seamless user experience by making the chatbot easily accessible within the familiar LMS environment. This integration eliminates the need for students to switch between platforms to seek help, thereby enhancing ease of access and maintaining continuity in their learning experience. By operating within Moodle's existing structure, MoodleBot interacts with students in a consistent and familiar setting, which is essential for increasing user engagement and reducing potential barriers to accessing assistance.

3. LangChain and LLM-Powered Agent:

   LangChain serves as the core framework that connects user inputs to the large language model (LLM) functionalities. Through LangChain, MoodleBot can handle complex, natural language queries by transforming them into vectors, allowing for accurate retrieval from the database.

- Answer Generation Tool: This tool is designed to access relevant course materials within the Weaviate vector database and retrieve the most relevant information for a given question. By retrieving specific, contextually aligned text chunks, MoodleBot can generate responses that are both informative and aligned with the course.
- Question Generation Tool: However, this tool creates practice questions for students. It can remind and generate review questions that support exam preparation, while drawing from course material. Based on flashcard-like activities, these practice questions simulate these practice, which are helpful for self regulated learning.

4. Retrieval-Augmented Generation (RAG):

- Combining Retrieval and Generation: GPT-4 with RAG is combining the vector database based content retrieval with generation in order to make sure that the generated response is grounded in actual and relevant material of the courses. The responses given by MoodleBot thus, are precise since the responses are specific on actual course content. [8]
- Iterative Refinement Process: MoodleBot does an iterative search for each MoodleBot query of the man education to the Weaviate database and fetch the best structured knowledge. If the reply it did send is lacking enough context, MoodleBot will keep looking for more information and will refine its answer when it finds it. Nevertheless, it allows the LLM to locate the right content in order to generate the desired response as complete and accurate as possible.
- Response Coherence and Accuracy: Retrieval and generation ensure correct responses from MoodleBot coherent, appropriate to context, and based on actual lecture text. It also reduces the likelihood the LLM will hallucinate answers in the iterative retrieval process.

5. Evaluation through Technology Acceptance Model (TAM):
The performance and users' satisfaction in MoodleBot are measured through Technology Acceptance Model, which identifies the perceived usefulness, perceived ease of use, and behavioral intention by the students toward the use of the chatbot. A to-

tal of forty-six students from the course of databases and information systems were selected, out of whom thirty completed the TAM questionnaire. With their prior knowledge in the course, these students provided the informed feedback regarding MoodleBot's accuracy and usability.

6. Manual and Automated Fact-Checking:
   In order to validate response accuracy, MoodleBot's responses were scored using two approaches — manual and automated — to confirm that responses were contextually accurate, aligning with course material. The chatbot's responses were reviewed manually to verify their factual correctness by a teaching assistant (TA) who compared them to lecture notes as well as other materials of its course. MoodleBot also uses LangChain's LLMSummarization-CheckerChain to do automated fact checking to complement this. This is a scalable way to verify response accuracy, by checking the alignment of MoodleBot's responses with the vector database. In addition to this, the TA also manual reviews the automated fact checking to ensure accuracy validation.

2.3.2 Advantages

MoodleBot provides several key benefits in supporting students of higher education within learning environments. Since academic support is continuously and freely available to a student 24/7, MoodleBot allows students to find information and help beyond class hours, enabling the student to stay active and progress independently. This will be beneficial for students who like to learn at their own pace, as well as those needing reinforcement on specific ideas. Additionally, the integration with tools of SRL will provide students with the opportunity to set goals and monitor their learning, as well as receive timely help, which increases motivation as well as academic outcomes.

2.3.3 Disadvantages

MoodleBot has various drawbacks that limit its usage and usability and scalability. Although it correctly answered 88% of the questions it responded to, sometimes some errors generate wrong answers; therefore, there is a very strong need for improved fact-checking. The operational costs also remain very high, especially in case of advanced models such as GPT-4, making this widely not feasible. The chatbot also relies heavily on extensive data input; when there are holes in the course materials, responses could be incomplete or incorrect. Lastly, though very helpful, many students still prefer human tutors, a clear indicator of the suitability of MoodleBot more for supplemental and less for foundational purposes.

2.3.4 Results

The results of the evaluation indicate that MoodleBot made a high accuracy rate of delivering course aligned responses at 88%, though there is still room for improvement in minimizing occasional errors. Analysis based on TAM indicated that students found MoodleBot useful and easy to use, with high scores in perceived usefulness and perceived ease of use. Notably, the students appreciated its constant availability and the ability of the chatbot to support SRL by helping immediately. In contrast, a lot of respondents expressed a need for human tutors, which, therefore, seems that MoodleBot best serves as an auxiliary, rather than fully replacing traditional learning. With the integration of Moodle and SRL tools, MoodleBot successfully provides a private, supportive space for students to seek help, particularly for those who are uncomfortable with public inquiries. Continuous 24/7 support fosters engagement, motivation, and goal-setting behaviors, while also reducing repetitive administrative tasks for educators.[9][10]

2.4 Natural Language Generation Model for Mammography Reports Simulation

2.4.1 Introduction

The *NLG Model* is designed to generate synthetic mammography reports for augmenting small annotated medical datasets while ensuring patient privacy. It leverages a **conditional RNN-LSTM architecture** that produces realistic and clinically accurate reports, capturing the style and diagnostic detail required for medical text generation [11]

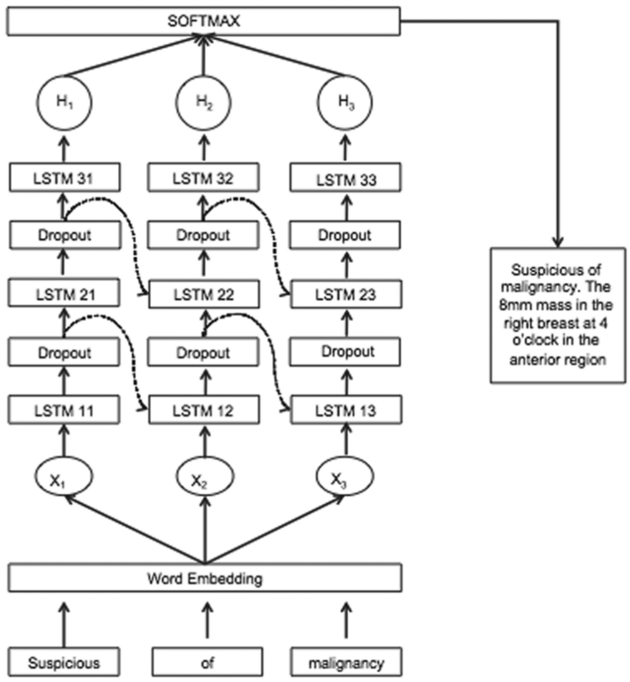


Figure 2.4: RNN-LSTM architecture for text-to-text report generation

2.4.2 Methodologies

- **Conditional RNN-LSTM Architecture**: Employs a recurrent neural network with LSTM cells, guided by specific diagnostic information to produce contextually relevant medical reports.
- **Word Embedding for Medical Terminology**: Incorporates specialized word embeddings that encode medical terminology to ensure accurate and relevant term usage in generated reports.
- **Data Augmentation through Synthetic Report Generation**: Generates additional labeled data by creating synthetic mammography reports, aiding in model training for downstream medical image classification tasks.
- **Privacy-Preserving Synthetic Data**: Ensures patient data privacy by generating reports that mimic real data without directly exposing patient details, making it suitable for sensitive applications.[12]

2.4.3 Advantages

- **Realistic Report Generation**: Produces highly realistic mammography reports, with 75% classified as real by radiologists.
- **Data Augmentation Benefits**: Enhances machine learning classification accuracy by providing additional training data through generated reports .
- **Better Performance Over Alternatives**: Outperforms Markov Random Field models in capturing clinical relevance and textual coherence.

2.4.4 Disadvantages

- **Model Complexity**: Requires significant computational resources due to its architecture, potentially limiting its practical application.
- **Generalization Limitations**: Model performance may be constrained to mammography and similar medical domains, limiting applicability across other fields.

2.4.5 Result

- **Improved Clinical Relevance**: The attention mechanism allows the model to focus on critical sections of medical data, resulting in synthetic reports that maintain high clinical relevance and accuracy.[13]
- **Enhanced Data Augmentation for Sensitive Fields**: By generating realistic medical reports, the model supports data augmentation, which is valuable in machine learning for healthcare, especially when real data is limited or sensitive.
- **Safeguarding Patient Privacy**: The model's synthetic data generation capabilities enable the creation of clinically accurate reports without using real patient data, offering a privacy-preserving solution for healthcare applications.
- **Adaptability to Complex Medical Terminology**: The attention-based RNN-LSTM framework efficiently handles complex medical terms and structures, enhancing the coherence and readability of generated reports for diverse healthcare contexts.

2.5 Summary and Gaps Identified

The table 2.1 provides an overview of the advantages and disadvantages of each paper discussed

Title	Advantages	Disadvantages
[1]Two-Step CNN Framework for Text Line Recognition in Camera-Captured Images	Efficient, on-device text recognition suitable for mobile systems; high accuracy and speed; privacy and latency benefits due to on-device processing.	Limited by mobile device computational constraints, could face issues with highly complex or blurred text from real-world scenarios.
[5] Neural Network-Based English Machine Translation System	Improved translation quality with reduced errors; BP neural network enables fluent and contextually relevant translations.	Heavy computational demands due to deep learning; translation performance highly dependent on neural network training data quality and volume.
[7]LLM-Powered Chatbot for Databases and Information Systems in Higher Education	High student acceptance; TAM model shows positive usability; accurate response generation using RAG and robust retrieval frameworks.	Fact-checking limitations affect response accuracy; some responses may still lack contextual relevance without continuous updates and refinement.
[11]Natural Language Generation Model for Mammography Reports Simulation	Effective data augmentation for limited medical datasets; RNN-LSTM generates coherent, realistic text for medical applications.	Limited to mammography reports; requires further validation and adaptation for other medical documentation styles.

Table 2.1: Summary of Key Research Works

Gaps Identified:

Despite all the progresses, several gaps remain in the current state of the art:

- Limited real-world adaptability: Models learned on specific datasets of educational input may fail for diverse input types including different fonts, languages, or technical diagrams..
- Computational Constraints: Applications like OCR, summarization, and multilingual support require significant processing power. Such a demand puts a constraint on performance on lower computational resources devices, making usability challenging in resource-limited environments.
- Fact-Checking and Consistency: An AI-driven chatbot would require improved mechanisms of fact checking to ensure accuracy, especially for complex or precise academic questions that are often important in an educational environment.
- Privacy and Data Security: Privacy and Data Security: Sensitive educational data requires strong privacy and security mechanisms, including data storage and controlled access, in order to secure user information.
- Scalability and Performance:With the potential for large volumes of data and concurrent users, the platform needs to remain responsive and maintain performance without degradation, which calls for an efficient, scalable architecture.
- Multilingual and Accessibility Challenges: High accuracy in translation of various linguistic structures is challenging. Support for a wide range of accents and dialects in speech-to-text functionality also remains challenging.

Chapter 3

System Design

3.1 System Architecture

The architecture diagram for the Synapse platform, presented in Fig. 3.1, illustrates how the combination of OCR, NLP, artificial intelligence chatbots, and multilingual integration works to allow for providing, transforming, and enriching educational materials for all end users.

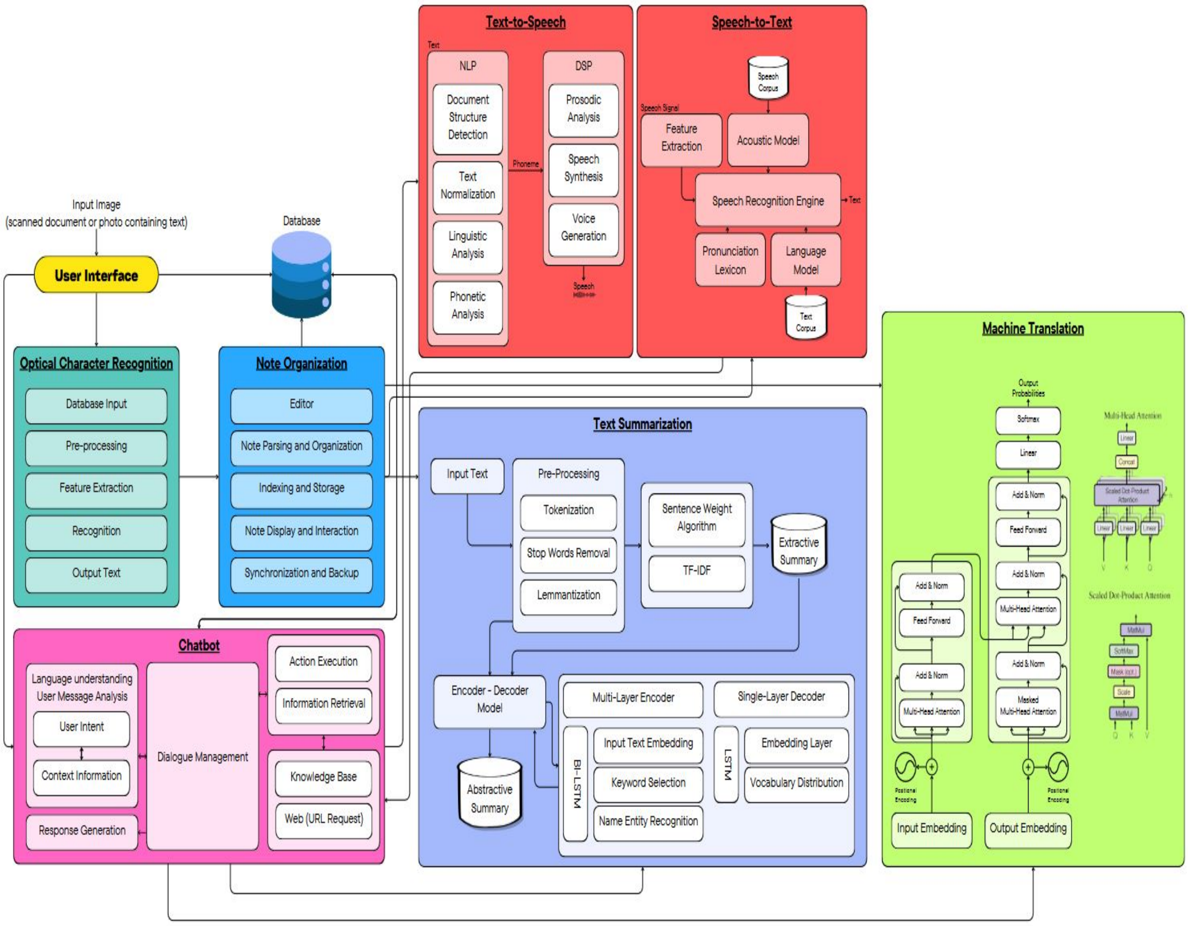


Figure 3.1: Synapse Architecture Diagram

3.2 Component Design

3.2.1 OCR and Text Extraction

This stage makes use of Optical Character Recognition for converting the text images into editable digital text. Preprocessing techniques, such as noise reduction, improve accuracy, allowing reliable text extraction from different types of educational materials. The OCR engine will be designed to support multiple languages, thereby meeting the demands of diverse content sources.

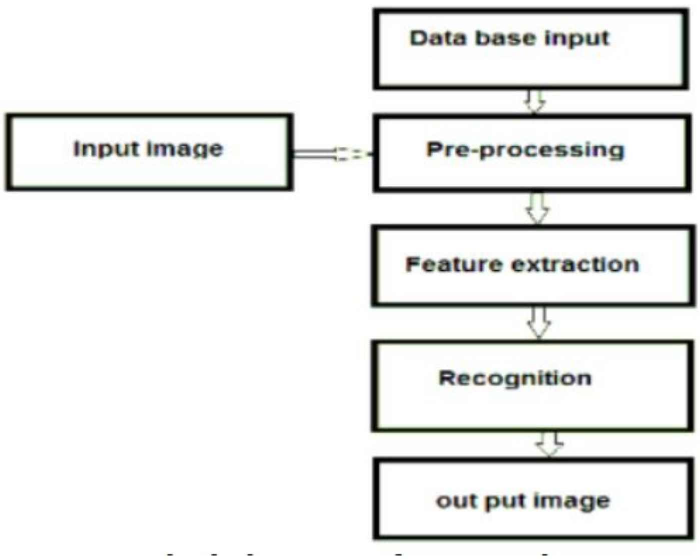


Figure 3.2: OCR Architecture Diagram

3.2.2 AI-Powered Chatbot

An AI-driven chatbot that interacts with users would help them learn better. Using Natural Language Processing (NLP), the chatbot will understand user questions and provide organized summaries, making the platform a personalized learning assistant..

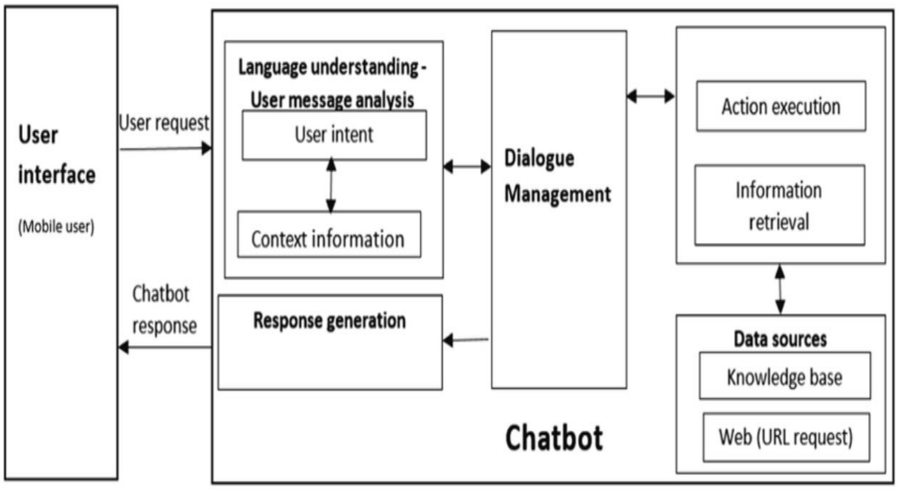


Figure 3.3: Chatbot Architecture Diagram

3.2.3 Text Summarization

This feature, through NLP techniques, summarizes long or complicated content into concise summaries so that the user can focus on the main points. This module enhances content comprehension and facilitates a quick review that supports efficient study and knowledge retention.

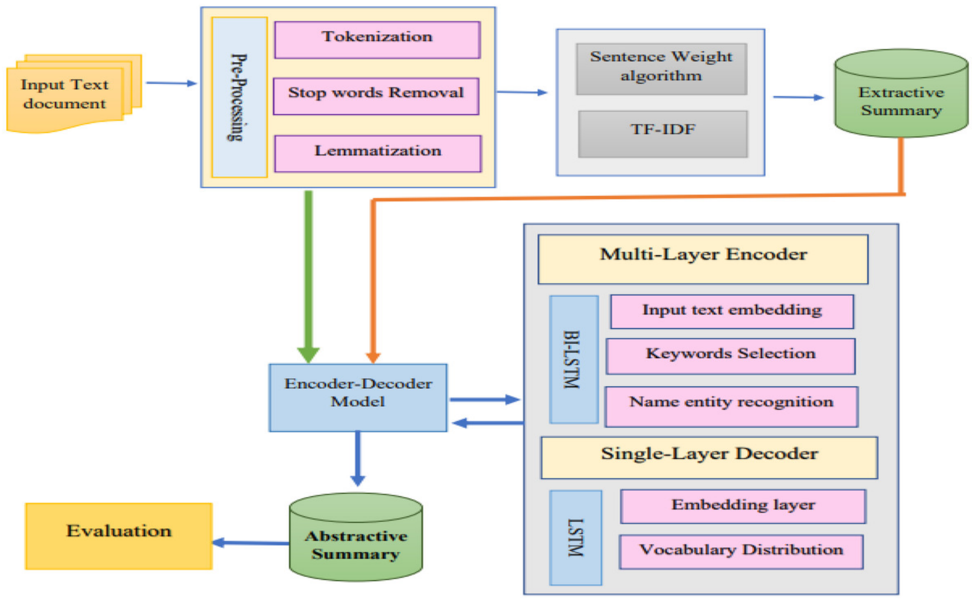


Figure 3.4: Text Summarization

3.2.4 Multilingual Support

Multilingual translation capabilities are implemented to ensure accessibility, thus allowing users to translate the content into multiple languages. The platform uses advanced NLP models, ensuring that educational materials are accessible to a diverse audience such that language barriers in learning are removed.

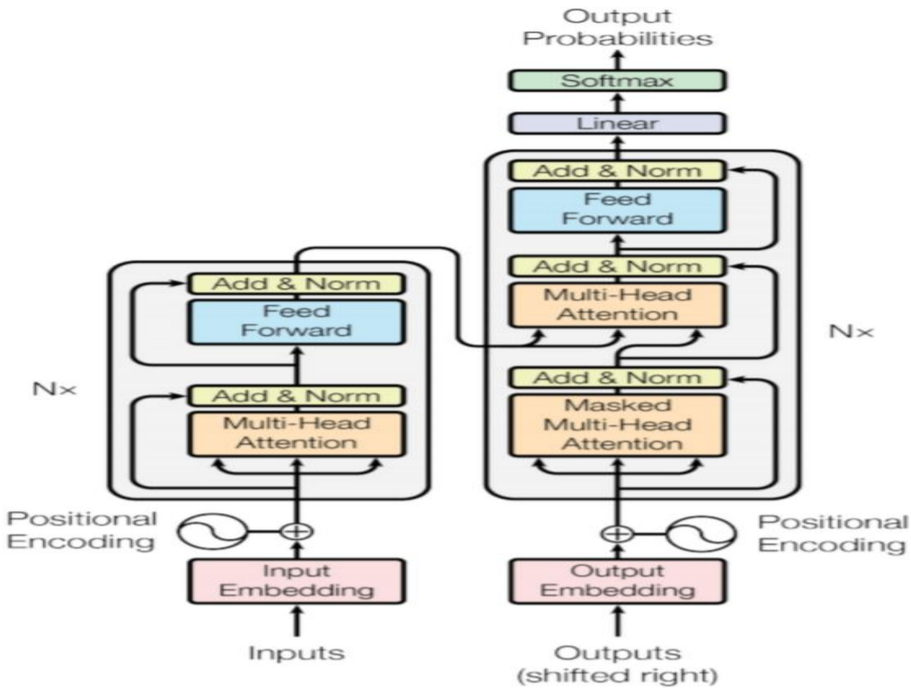


Figure 3.5: Transformer Model Architecture

3.2.5 Text-to-Speech

Text-to-speech converts the digital text into audio, helping in auditory learning and accessibility. This option ensure adaptation to different preferences for learning while promoting inclusiveness and accessibility.

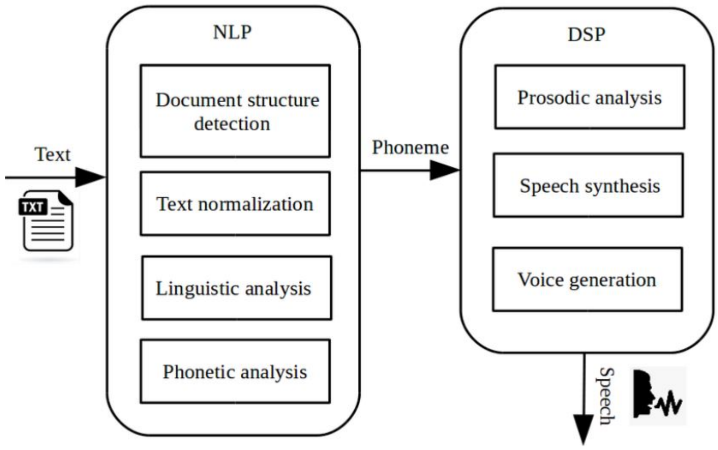


Figure 3.6: Text-to-Speech

3.3 Sequence Diagram

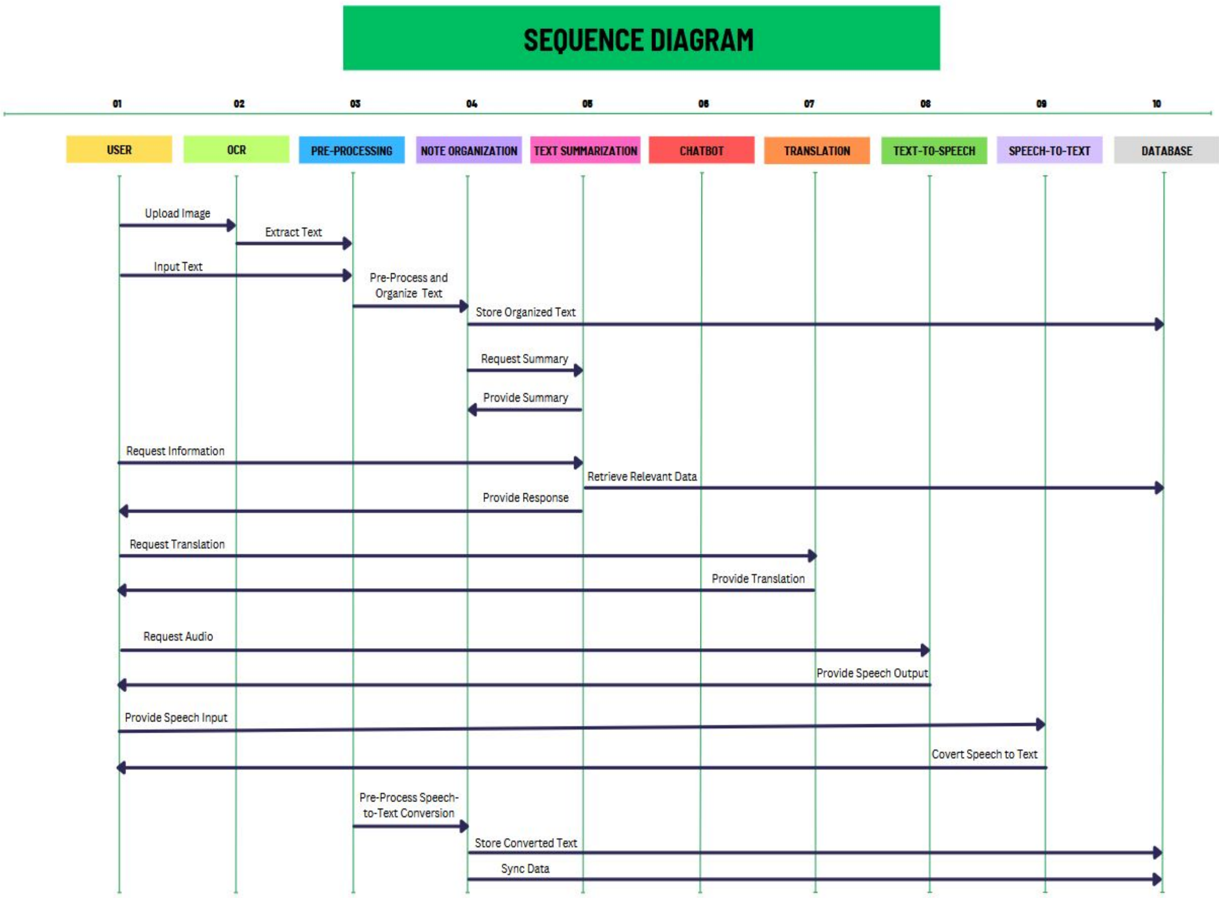


Figure 3.7: Sequence Diagram

3.4 Tools and Technologies

- **Software Requirements:**
  The project will use React.js adn Vite to build the frontend, while Python with FastAPI will be used for the backend. Reading and processing of images require Tesseract OCR, OpenCV, and Pillow (PIL). For language processing and machine learning, it will use spaCy, NLTK, and Transformers, and Dialogflow and Rasa will be used for the chatbot feature. For translation, Google Cloud or Microsoft Translator API are used. Speech functionalities like text-to-speech will use Google Cloud Text-to-Speech APIs.

- **Hardware Requirements:**
  The minimum hardware requirement for the Synapse is AMD Ryzen 5 or Intel Core i5, and it is recommended to use AMD Ryzen 7 or Intel Core i7 for better performance. Storage is provided by a 256 GB SSD; RAM must be at least 8 GB, but 16 GB is recommended for smoother operation. Supported operating systems are Linux, macOS, and Windows 10 or 11.

3.5 Module Divisions and work break down

3.5.1 Module Division

- **OCR and Text Extraction:** Focuses on converting images of text into editable and searchable digital formats using Optical Character Recognition (OCR) technology.

- **AI-Powered Chatbot:** Integrates an interactive chatbot powered by AI and NLP for answering questions, summarizing content, and assisting with note organization.

- **Text Summarization:** Utilizes NLP techniques to condense lengthy or complex educational materials into concise summaries.

- **Multilingual Support:** Provides translation services for multiple languages to enhance accessibility for a diverse user base.

- **Text-to-Speech:** Includes features for converting text to spoken language, supporting auditory and interactive learning styles.
- **User Interface Development:** Designs and implements a user-friendly and intuitive interface for seamless interaction with the platform.

3.5.2 Work Break Down

1. **OCR (Godwin Gino)**
   - OCR Development: Manage text extraction from images using advanced OCR techniques.
   - Frontend Development: Create the user interface for Synapse using React.js and React Native for a smooth and interactive user experience.
2. **Translation (Gautham C Sudheer)**
   - Translation Module: Build the multilingual translation feature to enhance content accessibility across various languages.
   - Backend Development: Design and maintain the backend system using Python and FastAPI to ensure efficient functionality.
3. **Chatbot (Fathima Jennath)**
   - Chatbot Development: Development of an AI chatbot to create an interface where users can be interacted with and which gives contextually correct responses.
   - Text Summarization: Creation of a module that will summarize content, and extract important points.
4. **Text-to-Speech (Mohammed Basil)**
   - Text-to-Speech Development: Develop a module that will convert text into natural-sounding speech.

3.6 Key Deliverables

The key deliverables of the Synapse project are that users can upload text images, which are converted into accurate, editable digital text for streamlined study and organization. An AI-powered chatbot will assist users in real time by answering questions, generating summaries, and guiding users through their notes in an interactive way. The platform will offer strong translation capabilities to make educational materials available in preferred languages, making the platform more inclusive. It will also enable effective interaction with text-to-speech functionality for auditory learning, thus ensuring accessibility and meeting different learning preferences.

3.7 Project Timeline

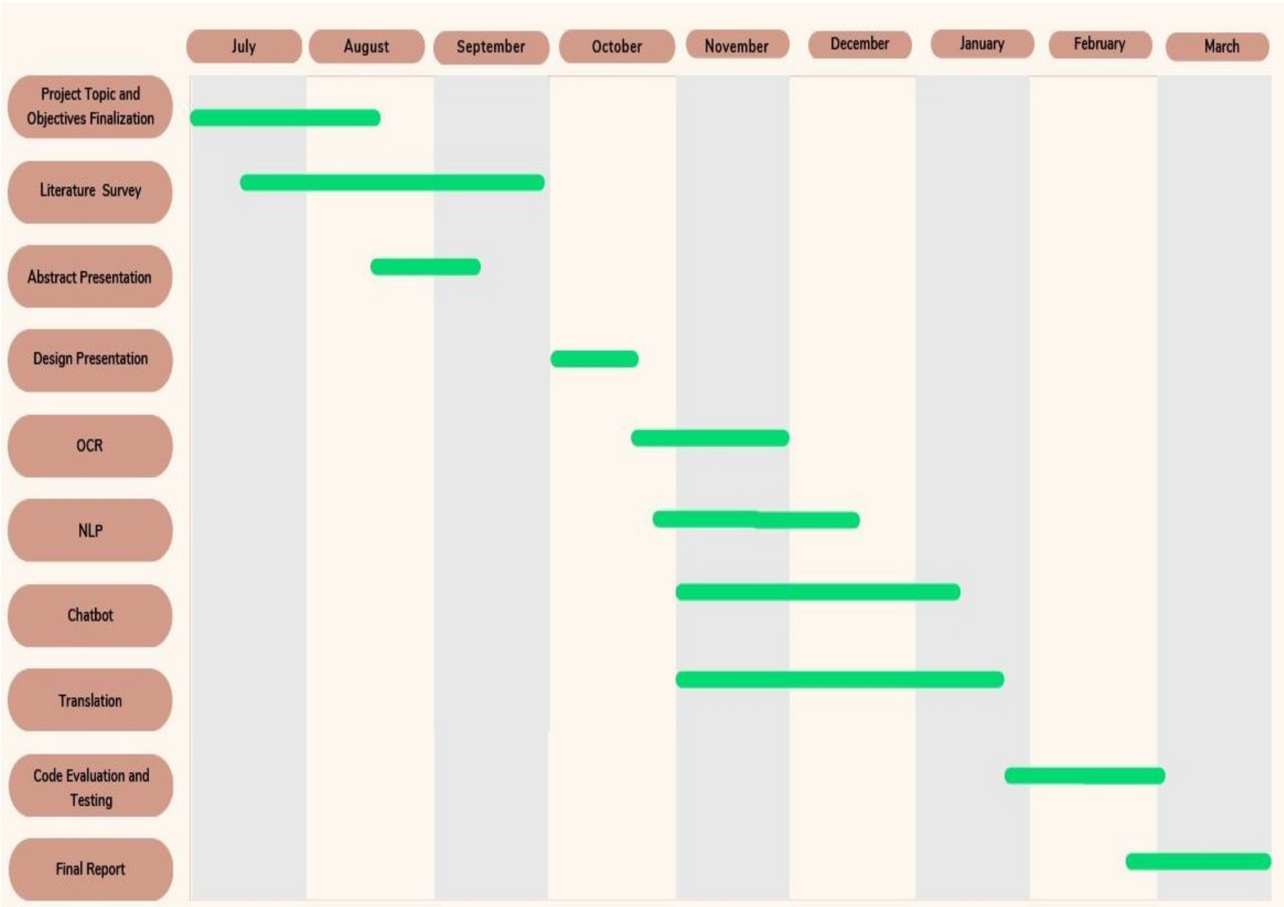


Figure 3.8: Gantt Chart

3.8 Conclusion

This chapter describes the system design for the Synapse project, including architecture, component design, tools, technologies, and requirements. It is built on a modular approach that includes features like OCR, AI chatbots, text summarization, multilingual support, and text-to-speech functionality, making it more accessible and interactive. This design complies with the idea of transforming educational resources into friendly digital formats.

Chapter 4

Implementation

4.1 Working of the System

Synapse is an AI-based learning platform that reads text from images and PDFs, facilitates chatbot-enabled learning, offers summarized insights, facilitates multilingual translation, and provides text-to-speech (TTS) functionality.

It combines Tesseract OCR (for extracting text), Gemini API (for interactive AI responses), Google Translate API (for multilingual support), and FastAPI (as the backend framework).

4.1.1 User Uploads an Image or PDF

- Users can upload image files (JPEG, PNG) or PDFs.
- If a PDF is uploaded:
  - It is converted into images using pdf2image.
  - Each page is saved as an image for further processing.
- The uploaded image or extracted PDF pages are sent for text extraction.

4.1.2 Text Extraction Using Tesseract OCR

Optical Character Recognition (OCR) is a core technology in digitization of documents, allowing to transform scanned copies or printed pages into machine-readable text. Several steps are implied in this transformation, such as image preprocessing, text recognition, and post-processing.

4.1.2.1 Image Preprocessing

To enhance text extraction accuracy, several preprocessing techniques are applied:

- **Grayscale Conversion:** Converts images to grayscale to enhance contrast and reduce noise. Given an image $I(x, y)$ with RGB values, grayscale conversion is performed using:

$$I_{gray}(x, y) = 0.299R + 0.587G + 0.114B \tag{4.1}$$

- **Noise Reduction:** Removes distortions using **Gaussian blur** or median filtering.

- **Binarization:** Applies **Otsu's thresholding**, which determines an optimal threshold T by minimizing intra-class variance:

$$T = \arg\min_{\tau} \left[\sigma_w^2(\tau)\right] \tag{4.2}$$

  where σ_w^2 is the weighted variance of foreground and background pixels.

- **Skew Correction:** Uses **Hough Line Transform** to detect and align tilted text.

4.1.2.2 Text Recognition with Tesseract OCR

- **Segmentation:** The image is broken down into logical units such as text blocks, lines, words, and characters to isolate meaningful content.

- **Character Recognition:** Tesseract uses a Long Short-Term Memory (LSTM)-based OCR engine, which employs deep learning techniques to match characters against trained datasets. This enables accurate recognition of both printed and handwritten text.

- **Multilingual Support:** Tesseract is trained on multiple languages, allowing it to recognize various scripts, fonts, and handwriting styles. This makes it adaptable to different types of documents, including multilingual texts.

In educational platforms like Synapse, OCR-based text extraction enhances accessibility, allowing users to search, summarize, and translate content. The integration of Natural Language Processing (NLP) and Machine Learning (ML) further enhances automated understanding and interaction with the text, benefiting both students and educators.

4.1.3 AI Chatbot for Interactive Learning (Gemini API)

The AI-powered chatbot in the Synapse platform is designed to provide interactive and intelligent assistance to users. It is built using advanced Natural Language Processing (NLP) and Machine Learning (ML) techniques, leveraging the Gemini API for efficient text understanding, conversation flow, and response generation. The chatbot's operation involves several stages, including text preprocessing, intent recognition, and response generation.

4.1.3.1 Text Preprocessing

Text preprocessing is the first stage in the chatbot's operation, which ensures that the raw input text is clean, consistent, and ready for analysis. The following tasks are performed during this phase:

- **Tokenization:** The input text is split into smaller units such as words, phrases, or sentences. Tokenization helps in understanding the structure of the sentence.
- **Normalization:** This step converts the text to a uniform format by transforming all text to lowercase, removing unnecessary punctuation, and correcting typographical errors.
- **Stopword Removal:** Common words such as "the," "is," and "and" are removed because they do not contribute meaningful information to the analysis.
- **Lemmatization:** Words are reduced to their base form (e.g., "running" becomes "run") to ensure that different forms of the same word are treated as a single entity.

These preprocessing tasks ensure that the chatbot processes only the relevant information, making it easier to derive meaning from the text.

4.1.3.2 Context Understanding

Once preprocessed, the text is analyzed to determine the user's intent and extract relevant information. This step involves:

- **Intent Recognition**: The user's queries are classified into predefined categories such as explanation, paraphrasing, or summarization. A supervised machine learning model or rule-based classification is used for this task.
- **Topic Extraction**:Key themes in the extracted text are identified using methods such as Term Frequency-Inverse Document Frequency (TF-IDF) or Latent Dirichlet Allocation (LDA).
- **Sentiment Analysis**: The emotional tone (positive, negative, or neutral) of the extracted text is identified to adapt response generation accordingly.

4.1.3.3 Response Generation

The chatbot employs the **Gemini API** for real-time response generation based on the analyzed text and user query. The process consists of:

1. **Context-aware Prompting**: The extracted text and user query are structured into a well-defined prompt for the Gemini API.
2. **API Processing**: The API processes the input and generates a coherent, contextually relevant response.
3. **Adaptive Response Formatting**: The response is refined based on the extracted content and query requirements.

4.1.4 Multilingual Support via Google Translate API

Multilingual support in the Synapse platform is implemented using the Google Translate API, enabling real-time translation of educational content to enhance accessibility and interaction for users from diverse linguistic backgrounds.

4.1.4.1 Text Input and Preprocessing

The platform accepts text input through various methods, such as manual user entry, OCR (for text extracted from images and PDFs), and speech-to-text. These methods are pre-processed to normalize text, remove irrelevant characters, and prepare the content for further processing. The pre-processing ensures that the text is clean, consistent, and ready for language detection and translation.

4.1.4.2 Language Detection

Google Translate API automatically detects the input language using machine learning algorithms trained on multilingual datasets. It analyzes n-gram patterns and character-level features, ensuring accurate detection for mixed-language content.

4.1.4.3 Translation Using Neural Machine Translation (NMT)

Once the source language is identified, Google Translate's Neural Machine Translation (NMT) processes the translation, ensuring context preservation through:

- **Encoding:** The input text is encoded into a high-dimensional vector representation, capturing the semantic meaning and structure of the sentence.
- **Contextual Translation:** Transformer-based models are employed to understand and translate text with consideration for contextual relationships between words, improving fluency and accuracy.
- **Attention Mechanism:** The attention mechanism focuses on the most relevant parts of the source sentence, improving translation accuracy for complex or longer sentences.
- **Decoding:** The encoded vector is decoded into the target language, generating a grammatically correct and contextually appropriate translation.

4.1.4.4 Post-Processing and Output Formatting

Once the translation is generated, the output undergoes post-processing to ensure it aligns with the platform's requirements. This step includes text formatting, special character handling, and adjustments to ensure seamless integration of the translated text into the user interface. The result is a fluently translated output that is presented to the user in the desired language.

4.1.5 Text-to-Speech (TTS)

The Text-to-Speech (TTS) functionality synthesizes speech from text. The process includes:

- **Text Input**: Preprocessing ensures correct formatting, handling abbreviations and symbols.
- **Speech Synthesis**: Google Cloud Text-to-Speech API, utilizing WaveNet technology, generates natural-sounding speech with pitch, rhythm, and intonation control.
- **Output**: The synthesized speech is delivered as an audio stream for playback through speakers or headphones.

4.1.6 FastAPI as the Backend Framework

1. FastAPI handles requests from the frontend.
2. It routes requests to:
    - OCR service for text extraction.
    - Gemini API for chatbot responses.
    - Google Translate API for multilingual support.
    - TTS API for speech processing.
3. FastAPI ensures asynchronous processing for fast responses.

4.2 Conclusion

This chapter includes AI-powered text extraction, chatbot learning, summarization, translation, and speech processing to enhance digital education. It covers the integration of Tesseract OCR, Gemini AI, Google Translate, and FastAPI to ensure an interactive, multilingual, and accessible learning experience.

Chapter 5

Results and Discussions

5.1 Results

The platform functioned effectively across all its core features, providing a seamless and user-friendly experience. Users were able to upload images and PDF files, from which the system accurately extracted content using Tesseract OCR. The chatbot, developed using the Gemini API, enabled users to engage with the extracted text, and the text-to-speech feature allowed users to listen to the content, supporting different learning styles. Significantly, all these major features are mainly supported with four languages: English, Malayalam, Spanish, and French. This multi-language support made the platform more inclusive in reaching users from different language backgrounds with ease. The following figures present the corresponding interface views that demonstrate each core feature of the platform.

5.1.1 OCR and Text Extraction

Optical Character Recognition (OCR) is used to extract text from images and PDFs. We compared different OCR systems based on accuracy, word error rate (WER), character error rate (CER), and precision to determine the most effective solution.

The dataset used for evaluation is the **IIIT 5K-word dataset**, containing 5,000 images of words from scene texts and digital images.

Comparison of OCR Systems

We tested AWS Textract, Tesseract, Google Cloud Vision, and Microsoft Azure OCR on large datasets.

OCR System	**Accuracy (%)**	**CER (%)**	**Remarks**
AWS Textract	99.3	1.3	Top performer when excluding outlier cases.
Tesseract	**98.9**	**3.6**	**Open-source; performs well with clear, high-contrast text.**
Google Vision OCR	98.0	5.3	High accuracy across various document types.
Microsoft Azure OCR	87.0	N/A	Struggles with handwritten and complex layouts.

Table 5.1: Performance of OCR Systems

Key Observations:

Based on Table 5.1, the following observations were made:

- **AWS Textract** had the highest accuracy (99.3%) and the lowest error rate.
- **Tesseract** (98.9%) performed well, especially for structured column-based text.
- **Google Vision** had a higher character error rate (5.3%).
- **Microsoft Azure** performed the worst, with only 87.0% accuracy.

Comparison of OCR Frameworks

Additionally, we evaluated Pytesseract, PyOCR, and EasyOCR as OCR frameworks.

OCR Framework	**WER (%)**	**CER (%)**	**Precision (%)**
Pytesseract	**16.7**	**3.6**	**92.3**
PyOCR	23.5	7.8	90.6
EasyOCR	34.3	19.4	83.4

Table 5.2: Performance of OCR Frameworks

Key Observations:

Based on Table 5.2, the following observations were made:

- **Pytesseract** had the highest precision (92.3%) and the lowest error rates.

- **PyOCR** had moderate performance but higher error rates.
- **EasyOCR** had the highest error rates (34.3% WER, 19.4% CER) and struggled with structured text.

AWS Textract is the most accurate commercial solution but is costly. Tesseract is a strong open-source alternative, especially when paired with Pytesseract, which offers the best precision among frameworks. Google Vision provides versatility but has higher error rates, while Microsoft Azure is the least effective for complex text.

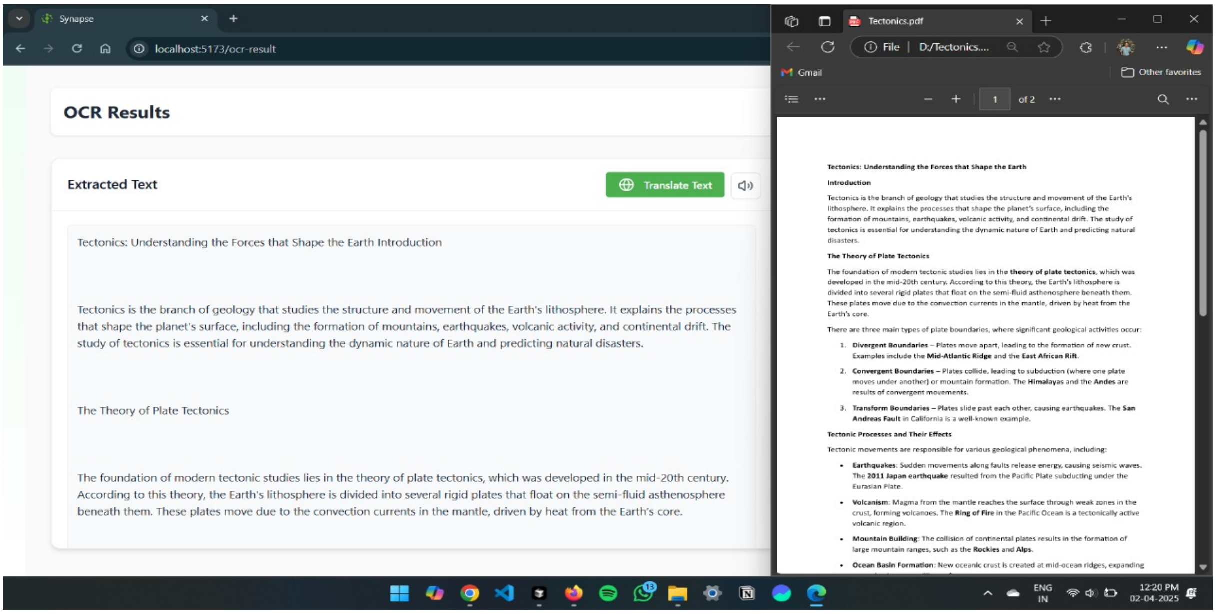


Figure 5.1: OCR and Text Extraction

5.1.2 Chatbot

We tested various chatbot models to evaluate their capabilities, knowledge performance, context window, and integration support. Our evaluation included three general-purpose chatbot models: Gemini, DeepSeek, and ChatGPT (GPT-4o). The comparison focused on their general knowledge understanding using the Massive Multitask Language Understanding (MMLU) benchmark, which assesses performance across multiple subjects, including mathematics, history, and computer science, as well as their context window size and multimodal capabilities.

Chatbot System	General Knowledge (MMLU)	Context Window	Multimodal Capabilities
Gemini	**90.0**	**Up to 1 million tokens**	**Text, images, video, voice**
DeepSeek	90.8	Up to 128K tokens	Text-based only
ChatGPT (GPT-4o)	88.3	Up to 128K tokens	Text and images

Table 5.3: Comparison of Chatbot Systems based on MMLU

Key Observations:

Based on Table 5.3, the following observations were made:

- **General Knowledge Performance (MMLU Score)**: DeepSeek achieved the highest score (90.8), followed by Gemini (90.0) and ChatGPT (GPT-4o) (88.3).
- **Context Window**: Gemini supports the largest context window, handling up to 1 million tokens, while both DeepSeek and ChatGPT (GPT-4o) support up to 128K tokens.
- **Multimodal Capabilities**: Gemini offers the most extensive multimodal support, handling text, images, video, and voice. ChatGPT (GPT-4o) supports text and images, while DeepSeek is limited to text-based interactions.

Additionally, summarization was performed using the Gemini API. This helped in condensing the evaluation results effectively while maintaining key details.

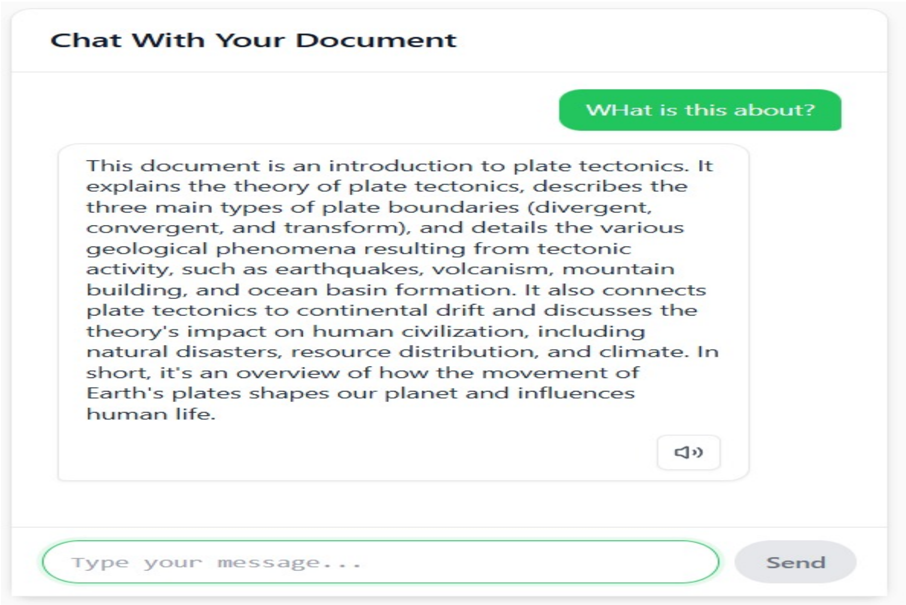


Figure 5.2: Chatbot System Response in English

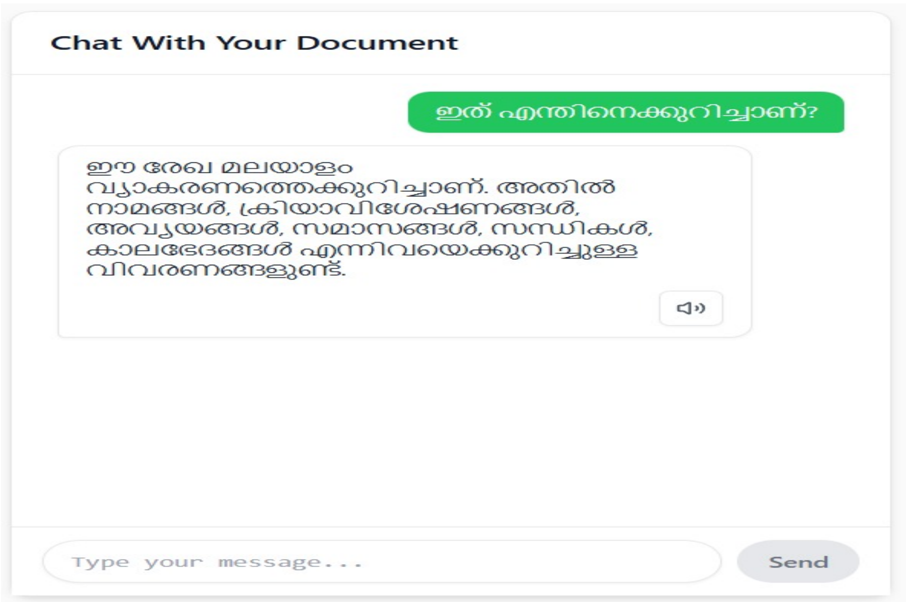


Figure 5.3: Chatbot System Response in Malayalam

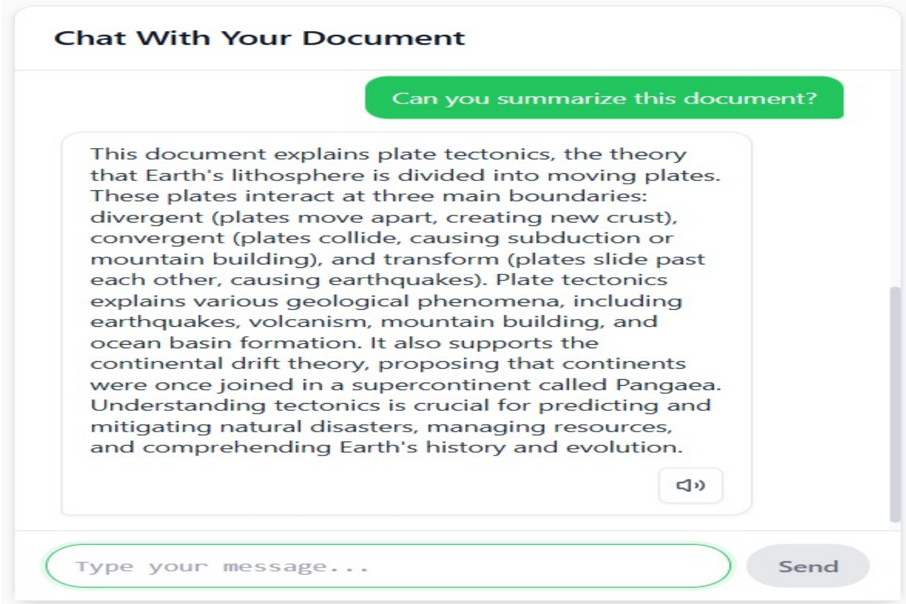


Figure 5.4: Summarization

5.1.3 Machine Translation

We evaluated three major machine translation techniques: Neural Machine Translation (NMT), Statistical Machine Translation (SMT), and Rule-Based Machine Translation (RBMT). The results are summarized in Table 5.4.

Approach	**Domain**	**BLEU Score**	**Accuracy**	**Notes**
NMT	**Spoken**	**0.4273**	**90.42%**	**Captures broader context; requires substantial data and computational resources.**
SMT	Spoken	0.3671	89.58%	Learns from bilingual corpora; may misinterpret context without sufficient data.
RBMT	Spoken	0.2570	84.17%	Effective with structured language; struggles with idiomatic expressions.

Table 5.4: Comparison of Machine Translation Approaches

Neural Machine Translation (NMT) performed best in terms of accuracy and BLEU score but requires significant computational resources. Statistical Machine Translation (SMT) showed decent accuracy but struggles with context misinterpretation when data is insufficient. Rule-Based Machine Translation (RBMT) was the least effective, particularly with idiomatic expressions, but works well with structured language.

Comparison of Translation Services

Different translation services were compared based on their translation quality, processing speed, and multilingual support. The results are shown in Table 5.5.

Translation Service	Translation Quality Score	Processing Speed (s)	Supported Languages
DeepL	8.38	0.51	31
Google Translate	**7.90**	**0.22**	**133**
Microsoft Translator	7.77	0.26	100+
Amazon Translate	N/A	0.33	75

Table 5.5: Comparison of Translation Services

DeepL had the highest translation quality score but supports fewer languages. Google Translate had a competitive translation quality score and the fastest processing speed while supporting the most languages (133). Microsoft Translator provides extensive language support (100+ languages) but has slightly slower processing than Google Translate. Amazon Translate had the slowest processing speed and does not provide an average quality score.

Neural Machine Translation is the most effective approach, especially for spoken language. Among translation services, Google Translate provides the best balance between translation quality, speed, and multilingual support, while DeepL offers higher translation quality in limited language pairs.

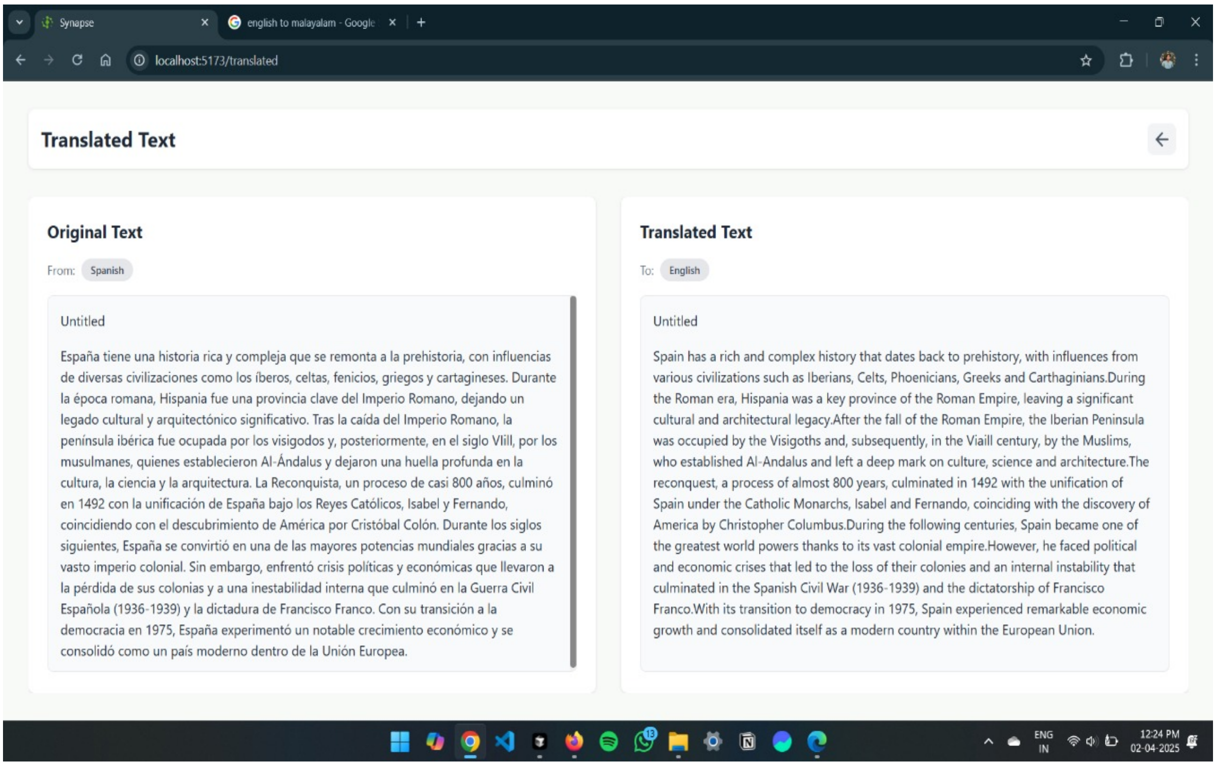


Figure 5.5: Translation

5.1.4 Text to Speech

We conducted a comparative analysis of various TTS platforms based on their language support, voice options, pricing, and key features.

Comparison of TTS Services

The table 5.6 compares popular TTS services, highlighting their capabilities and pricing structures.

Service	Languages	Voice Options	Pricing	Key Features
Google Cloud Text-to-Speech	**50+**	**Multiple**	**Usage-based**	**Customizable voices, Google Cloud integration**
Microsoft Azure Text-to-Speech	140+	Multiple	Usage-based	Customizable voices, Azure integration
Amazon Polly	39+	Multiple	Usage-based	Realistic speech, AWS integration
Synthesia	130+	Multiple	$22/month	Video editing, dubbing, subtitles, transcription

Table 5.6: Comparison of TTS Services

Key Observations

- **Language Support:** Azure (140+) leads, followed by Synthesia (130+).
- **Pricing:** Google Cloud, Azure, and Polly use usage-based pricing; Synthesia costs $22/month.
- **Features:**
  - Azure & Google Cloud: Seamless cloud integration.
  - Amazon Polly: Realistic speech generation.
  - Synthesia: Video editing, dubbing, and transcription.

5.2 Discussion

The system worked seamlessly on all its core functionalities—accurately extracting clear text, engaging users in insightful conversations across multiple languages, and enhancing accessibility via text-to-speech and translation features. The platform currently supports four languages: English, Malayalam, Spanish, and French, and is designed to accommodate more in the future. Although speech-to-text functionality has not yet been integrated, it remains a key focus for future development. In general, the interface was intuitive and responsive, allowing smooth transitions between features and languages, making the platform both effective and user-friendly for educational purposes.

The system was effective for the core functionalities such as accurately extracting text content, engaging users in insightful conversations across multiple languages, and improving accessibility through text-to-speech and translation capabilities. Currently the platform supports four languages: English, Malayalam, Spanish, and French, and other languages can be integrated in the future. Although the speech-to-text feature is not yet included, it continues to be one of the major areas of future development. As a whole, the interface is responsive and user-friendly, and it provides effortless transitions between languages and features, making the platform both effective and easy to use for educational objectives.

5.3 Conclusion

This chapter includes an evaluation of OCR, chatbot, and translation systems. It reviews the project's results and discussions, demonstrating the successful implementation of key functionalities such as efficient text extraction, multilingual communication, and enhanced accessibility. Additionally, it explores potential enhancements, including the future inclusion of speech-to-text functionality, and highlights the project's potential for future development.

Chapter 6

Conclusion

6.1 Conclusion

The project Synapse: Where AI Meets Education successfully demonstrates how artificial intelligence can be meaningfully applied to enhance the educational experience. By incorporating fundamental features like image and PDF upload, OCR-based text extraction, intelligent chatbot interaction, text-to-speech, and multilingual support, the platform offers an integrated learning environment that addresses diverse user requirements and interests. The support for English, Malayalam, Spanish, and French in the system enhances inclusivity, enabling users from various linguistic backgrounds to access educational content more effectively .

While the platform already generates strong outcomes with its existing features, it has also been developed for scalability. Future possibilities include the implementation of speech-to-text capability, wider language coverage, and continuous optimization of AI-based capabilities to enable adaptive learning as well as content customization. The success of this project highlights the growing potential of AI to revolutionize how education is created and consumed. As the education sector continues to evolve, tools like Synapse present the possibility for more intelligent, accessible, and inclusive learning platforms that benefit students and teachers in achieving improved results through technology.

6.2 Future Scope

Synapse can further improve by incorporating more features that enhance usability and precision. One significant development area is the incorporation of handwritten text recognition using deep learning techniques such as Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs). This will allow the system to extract text from handwritten class notes and scanned academic publications, significantly broadening its usage in real educational environments. Also, the intended integration of speech-to-text capability will create a completely interactive voice interface, making the platform more convenient for users with diverse needs, such as those with visual or motor disabilities.

Language support extension is another fundamental area that will enhance the global coverage and inclusivity of the system. Beyond the existing four languages of English, Malayalam, Spanish, and French, more users will be able to access content in their own language, leading to better engagement. AI-driven adaptive learning capabilities can adapt content to the individual learner's style and pace. Moreover, integration with Learning Management Systems (LMS) will make the platform compatible with institutional requirements. Such developments will place Synapse as a rich, smart, and accessible teaching aid for all kinds of learning environments.

References

[1] Y. S. Chernyshova, A. V. Sheshkus, and V. V. Arlazarov, "Two-step cnn framework for text line recognition in camera-captured images," *IEEE Access*, vol. 8, pp. 32 587–32 599, 2020.

[2] P. Kumar, "A review on cnn and its applications in image processing," *Journal of Advances in Mathematics and Computer Science*, vol. 29, no. 5, pp. 1–9, 2019.

[3] J. Wang, "A study of the ocr development history and directions of development," *Highlights in Science, Engineering, and Technology*, vol. 72, pp. 409–415, 2023, licensed under CC BY-NC 4.0.

[4] S. Ayyalasomayajula and S. S. Kolluri, "Cnn-based optical character recognition and applications," in *IEEE International Conference on Intelligent Sustainable Systems (ICISS)*. IEEE, 2017.

[5] S. Lei and Y. Li, "English machine translation system based on neural network," *Procedia Computer Science*, vol. 228, pp. 409–420, 2023.

[6] R. Dabre, T. Nakazawa, and A. Kunchukuttan, "A voyage on neural machine translation for indic languages," in *Proceedings of the 6th International Conference on Asian Language Processing (IALP)*. IEEE, 2018.

[7] A. T. Neumann, Y. Yin, S. Sowe, S. Decker, and M. Jarke, "An llm-driven chatbot in higher education for databases and information systems," *IEEE Transactions on Education*, 2024.

[8] OpenAI, "GPT-4 Technical Report," Online, 2023, available: https://cdn.openai.com/papers/gpt-4.pdf.

[9] S. Yu, Y. Chen, and H. Zaidi, "Ava: A financial service chatbot based on deep bidirectional transformers," *Frontiers in Applied Mathematics and Statistics*, vol. 7, 2021.

[10] A. K. Surendran, R. Murali, and R. K. Babu, "Conversational ai - a retrieval based chatbot," *EasyChair Preprint*, no. 4020, August 2020.

[11] A. Hoogi, A. Mishra, F. Gimenez, J. Dong, and D. Rubin, "Natural language generation model for mammography reports simulation," *IEEE Journal of Biomedical and Health Informatics*, vol. 24, no. 9, pp. 2711–2721, 2020.

[12] C. C. Aggarwal, "On k-anonymity and the curse of dimensionality," in *Proceedings of the 31st International Conference on Very Large Data Bases*. VLDB Endowment, 2005, pp. 901–909.

[13] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *arXiv preprint arXiv:1409.0473*, 2015.

Appendix A: Presentation

FINAL PRESENTATION

Synapse

WHERE AI MEETS EDUCATION

Dr. Mary Priya Sebastian

Gautham C Sudheer
Fathima Jennath N K
Godwin Gino
Mohammed Basil

CONTENT

Problem Definition

The traditional process of converting physical text into digital, editable, and searchable formats is often time-consuming and inefficient, particularly for students and educators working with notes or multilingual content.

Purpose and Need

The project aims to meet the growing demand for digitized, interactive educational content by leveraging AI-powered tools.

- **Enhance Accessibility**: Converting text images, and multilingual content, into editable and searchable formats
- **Streamline Learning and Note-Taking**: Digitize and organize notes, to interact with content through features like AI-powered chatbots.
- **Promote Inclusive and Interactive Education**: With multilingual support and speech-to-text/text-to-speech functionalities

Objectives

01 **Develop an AI-powered Platform For Text Digitization**
Efficiently converts text images into editable and searchable digital content using advanced OCR technology.

02 **Integrate Natural Language Processing (NLP) Capabilities**
Enable interactive features, such as an AI chatbot for answering questions, generating summaries, and assisting with note organization.

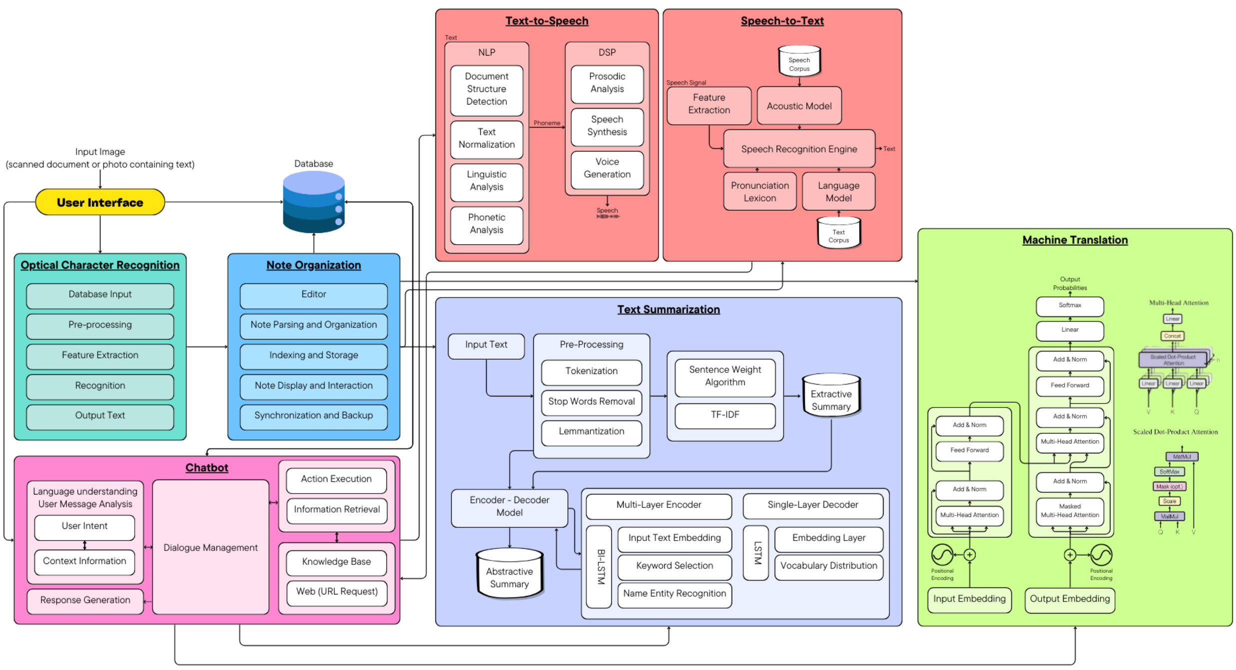
03 **Provide Multilingual Support and Accessibility Features**
Speech-to-Text and Text-to-Speech, ensuring the platform is adaptable to users from diverse linguistic backgrounds and varying learning needs.

Literature Survey

Paper	Advantages	Disadvantages
S. Lei and Y. Li, "English Machine Translation System Based on Neural Network Algorithm," Procedia Computer Science, vol. 228, pp. 409-420, 2023	Better handling of long-range dependencies, parallel processing for faster training	High computational and memory requirements, for long sequences, and they can be data-hungry
A. T. Neumann, Y. Yin, S. Sowe, S. Decker, and M. Jarke, "An LLM-driven chatbot in higher education for databases and information systems," IEEE Transactions on Education, vol. 1, pp. 1-15, 2024	Provides personalized, quick responses to help students understand course material	The chatbot sometimes gives incorrect or repetitive answers, meaning it still needs better fact-checking to ensure accuracy
N. Sarika, N. Sirisala, and M. S. Velpuru, “Cnn based optical character recognition and applications,” Proc. Sixth Int. Conf. Inventive Comput. Technol. (ICICT 2021), pp. 666-672, 2021	VGG-16 model has shown 92% accuracy in Telugu handwritten character recognition	It requires more computational resources and longer training times, making it less efficient for scenarios with limited computational power.

Literature Survey

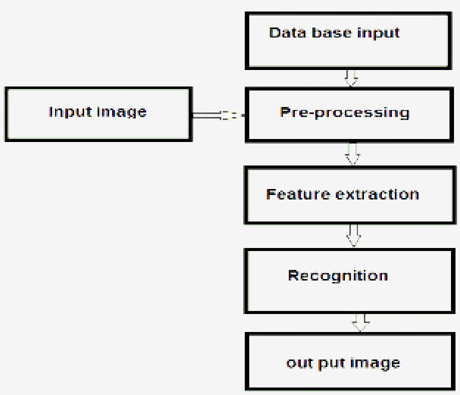
Paper	Advantages	Disadvantages
Rayyan Najam and Safiullah Faizullah, “Analysis of Recent Deep Learning Techniques for Arabic Handwritten-Text OCR and Post-OCR Correction” Applied Sciences, vol. 13, no. 13, p. 7568, Jun. 2023	Advanced architectures like Transformer-based models and RNNs can capture contextual relationships between characters, words, and lines	Deep learning models require large labeled datasets of handwritten text to achieve high accuracy
Lorenz Kuhn, Yarin Gal, Sebastian Farquhar, “Semantic Uncertainty: Linguistic Invariances for Uncertainty Estimation in Natural Language Generation” ICLR 2023, https://doi.org/10.48550/arXiv.2302.09664	Semantic Entropy captures uncertainty over meanings, not just forms, which provides a more reliable measure in tasks like question answering	The bidirectional entailment algorithm for clustering sentences has quadratic complexity; The performance is highly sensitive to sampling temperature and methods



MODULE DESCRIPTION

1. OCR & Text Extraction

- Build a microservice that handles image uploads, processes them with OCR, and returns editable text.
- Image processing techniques and deep learning models to recognize and extract text accurately



1. OCR & Text Extraction

- Preprocessing (Enhancing Image Quality)
  - Grayscale Conversion: Reduces complexity, improves contrast.
  - Binarization (Otsu's Thresholding): Converts grayscale to binary for better text separation.
  - Noise Reduction: Gaussian blur/median filters remove distortions.
  - Skew Correction (Hough Transform): Detects & corrects text misalignment.

1. OCR & Text Extraction

- Text Detection
  - EAST (Efficient & Accurate Scene Text Detector): Uses deep learning to locate text regions in images.
- Text Recognition
  - CRNN (Convolutional Recurrent Neural Network):
    - CNN extracts features.
    - LSTM models sequential text for accurate OCR.

OCR System	Accuracy (%)	CER (%)	Remarks
AWS Textract	99.3	1.3	Top performer when excluding outlier cases.
Tesseract	98.9	3.6	Open-source; performs well with clear, high-contrast text.
Google Cloud Vision OCR	98.0	5.3	High accuracy across various document types.
Microsoft Azure OCR	87.0	N/A	Struggles with handwritten and complex layouts.

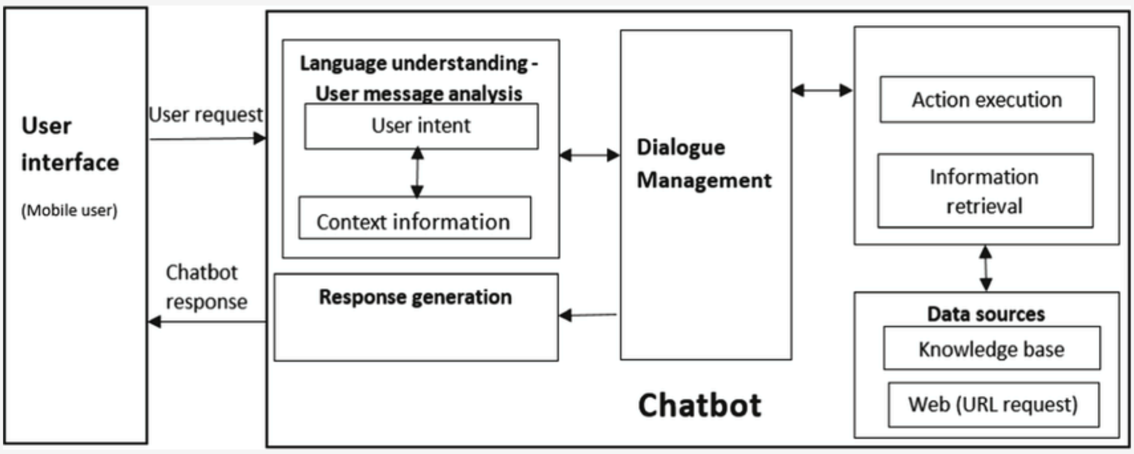
Evaluated using IIIT 5K-word Dataset, Contains 5,000 images of cropped word instances from scene texts and born-digital images, each associated with a 50-word and a 1,000-word lexicon.

OCR Framework	Word Error Rate (%)	Character Error Rate (%)	Precision (%)
Pytesseract	16.7	3.6	92.3
PyOCR	23.5	7.8	90.6
EasyOCR	34.3	19.4	83.4

2. AI-Powered Chatbot

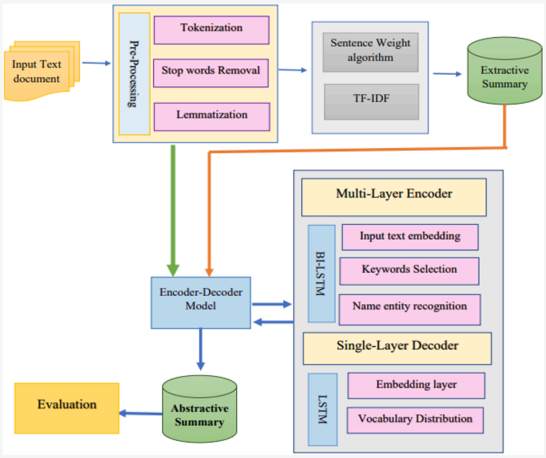
- Natural Language Understanding (NLU)
  - Tokenization & Named Entity Recognition (NER): Identifies key entities using NLP tools (SpaCy, Hugging Face).
  - BERT/DistilBERT: Captures context for better query understanding.
- Intent Recognition
  - Classification Model (BERT/RNN-based): Determines user intent.
  - Entity & Context Extraction: Ensures precise responses.
- Response Generation
  - GPT/Dialogflow: Generates context-aware replies.

2. AI-Powered Chatbot



3. Text Summarization

- Use pre-trained models from Hugging Face Transformers.
- Implement a summarization API that can be called from the frontend.



3. Text Summarization

- Preprocessing
  - Tokenization: Splits text into words & sentences.
  - Lemmatization & Stopword Removal: Reduces words to base forms & removes irrelevant words.
- Feature Representation
  - Word Embeddings (Word2Vec, GloVe): Captures word meanings.
  - Contextual Embeddings (BERT): Understands word meaning in context.

Chatbot System	General Knowledge (MMLU)	Context Window	Multimodal Capabilities
Gemini	90.0	Up to 1 million tokens	Text, images, video, voice
DeepSeek	90.8	Up to 128K tokens	Text-based only
ChatGPT (GPT-4o)	88.3	Up to 128K tokens	Text and images

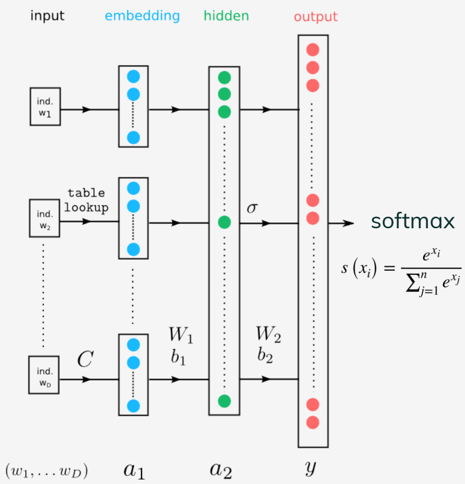
Evaluated using the Massive Multitask Language Understanding (MMLU) benchmark. MMLU comprises approximately 16,000 multiple-choice questions spanning 57 subjects, including mathematics, history, computer science, law, and more.

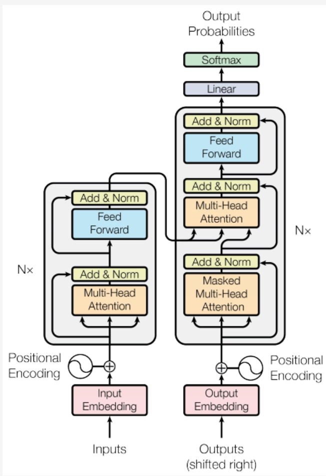
Chatbot System	Natural Language Understanding (NLU)	Integration Capabilities	Pricing	Supported Languages
Amazon Lex	Yes	AWS services; platforms like Facebook Messenger, Slack, Twilio, Kik	$0.004 per speech request; $0.00075 per text request	Primarily English
Google Chat API	No	Google Workspace services; enhances Google Chat functionalities	Varies based on usage; aligned with Google Cloud's pricing models	Multiple languages, depending on implementation
Meta Messenger API	No	Facebook Messenger; connects to other services via webhooks and APIs	Varies based on usage;	Multiple languages, depending on implementation

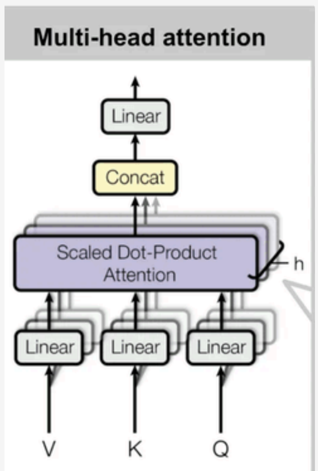
4. Multilingual Support

- Preprocessing
  - Tokenization & Sentence Splitting: Prepares text for translation.
- Translation Model
  - Transformer-based Models: Leverages multilingual datasets.
  - Attention Mechanism: Aligns & generates context-aware translations.
- Optimization
  - Fine-tuning: Enhances accuracy for specific languages & domains.

4. Multilingual Support







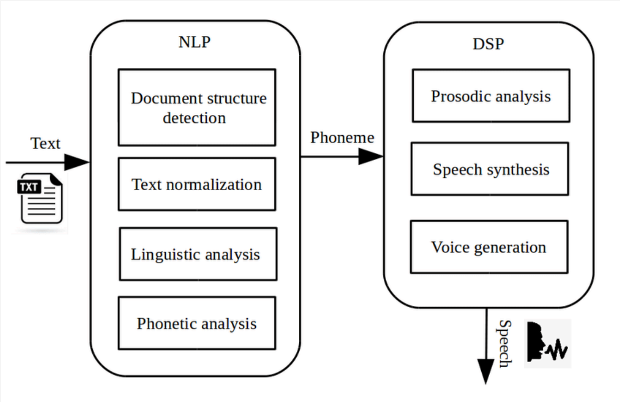
Approach	Domain	BLEU Score	Accuracy	Notes
NMT	Spoken	0.4273	90.42%	Captures broader context; requires substantial data and computational resources.
SMT	Spoken	0.3671	89.58%	Learns from bilingual corpora; may misinterpret context without sufficient data.
RBMT	Spoken	0.2570	84.17%	Effective with structured language; struggles with idiomatic expressions.

Translation Service	Average Translation Quality Score	Processing Speed (seconds)	Supported Languages
DeepL	8.38	0.51	31
Google Translate	7.90	0.22	133
Microsoft Translator	7.77	0.26	100+
Amazon Translate	N/A	0.33	75

Evaluated using WMT (Workshop on Machine Translation) Datasets, and IWSLT (International Workshop on Spoken Language Translation) Datasets, annual datasets provided by the WMT and IWSLT conference for shared tasks in MT, covering multiple language pairs and domains.

5. Text-to-Speech

- Provide options for different languages and voices
- Use Google Text-to-Speech API for reading out notes or chatbot responses



5. Text-to-Speech

- Preprocessing
  - Text Normalization: Converts text into phonetic form (handles abbreviations, numbers, symbols).
- Linguistic Processing
  - Prosody Analysis: NLP models determine intonation, rhythm, and stress for natural speech.

5. Text-to-Speech

- Speech Synthesis
  - Tacotron 2 / FastSpeech: Converts phonetic sequences into spectrograms.
  - Neural Vocoder (WaveNet / HiFi-GAN): Generates high-quality speech from spectrograms.

Service	Number of Languages	Voice Options	Pricing	Key Features
Google Cloud Text-to-Speech	50+	Multiple voices	Usage-based pricing	Customizable voices, integration with Google Cloud services
Microsoft Azure Text to Speech	140+	Multiple voices	Usage-based pricing	Customizable voices, integration with Azure services
Amazon Polly	39+	Multiple voices	Usage-based pricing	based pricing Realistic speech generation, integration with AWS services
Synthesia	130+	Multiple voices	$22 per month	Video editor, dubbing, subtitles, transcription

Comparing GCP & AWS NLP Services

Service	Feature	Google Cloud	AWS
OCR	API	Cloud Vision API	Amazon Textract
	Capabilities	Text detection, handwriting recognition, multi-language support	Text extraction, table and form data extraction, handwriting recognition
	Supported Formats	Images (JPEG, PNG, GIF)	Documents (PDF, TIFF), Images (JPEG, PNG)
	Pricing	Based on image count; details at Google Cloud Vision Pricing	Based on pages processed; details at Amazon Textract Pricing

Service	Feature	Google Cloud	AWS
Machine Translation	API	Translation API	Amazon Translate
	Supported Languages	100+	75
	Customization	Glossary support	Custom terminology
	Pricing	$20 per million characters; details at Google Cloud Translation Pricing	$15 per million characters; details at Amazon Translate Pricing

Service	Feature	Google Cloud	AWS
Text-to-Speech (TTS)	API	Text-to-Speech	Amazon Polly
	Languages & Voices	50+ languages, multiple voices	41 language variants, 100+ voices
	Features	Voice cloning, pitch and speed control	Per-word timestamps, pitch and speed control
	Pricing	Free tier: 1M characters/month; Standard: $16 per 1M characters; details at Google Cloud Text-to-Speech Pricing	Free tier: 1M characters/month for 12 months; Standard: $16 per 1M characters; details at Amazon Polly Pricing

Assumptions

- Users are expected to provide clear, high-resolution images for the OCR module to effectively extract text
- Access to reliable language models and translation algorithms that can handle a wide variety of languages and dialects with minimal errors
- A consistent and stable internet connection is assumed for smooth interaction with the backend and AI modules
- The AI-powered chatbot and NLP systems are expected to improve with continued use, requiring regular updates and model training

Requirements

- Frontend Development: React.js and React Native
- Backend Development: Python and Django
- OCR and Image Processing: Tesseract OCR, OpenCV, and Pillow (PIL)
- NLP and Machine Learning: spaCy, NLTK, and Transformers
- Chatbot and Conversational AI: Dialogflow and Rasa
- Translation Services: Google Cloud and/or Microsoft Translator API
- Speech Processing: Google Cloud Speech-to-Text API and Text-to-Speech API

Requirements

- Processor: Intel Core i5 or AMD Ryzen 5 (minimum); Intel Core i7 or AMD Ryzen 7 (recommended)
- RAM: 8 GB (minimum); 16 GB or more (recommended)
- Storage: 256 GB SSD (minimum); 512 GB SSD or higher (recommended)
- Graphics Card: Integrated graphics are sufficient for development; dedicated GPU recommended for intensive tasks like machine learning.
- Operating System: Windows 10 or 11, macOS, or a Linux distribution.

Risk and Challenges

- Accurately extracting text from diverse input types and understanding technical language, leading to unreliable outputs.
- Ensuring high translation accuracy across various languages can be challenging, especially with diverse linguistic structures.
- Efficiently handle high volumes of data and concurrent requests without performance degradation, requiring a scalable architecture.
- AI-powered features might demand significant computational resources, leading to potential high operational costs.

Expected Output

- Users can upload text images and receive accurate, editable digital text for easier study and organization.
- An AI-powered chatbot providing real-time assistance by answering questions, generating summaries, and helping users navigate their notes conversationally.
- Robust translation capabilities, enabling users to access educational materials in preferred languages, enhancing inclusivity.
- Utilize speech recognition for note-taking and text-to-speech functionality to listen to notes, promoting accessibility and varied learning methods.

Conclusion

"Synapse" is an AI-powered platform that transforms text images into editable digital content, enhancing the educational experience through advanced OCR, NLP techniques, and interactive features.

- Converts printed and handwritten content into editable digital formats, enhancing accessibility and organization
- Provides real-time, interactive management of notes through conversational AI, improving user engagement and efficiency.
- Text summarization, speech processing, and efficient note organization to support diverse learning needs and streamline educational practices.

Appendix B: Vision, Mission, Programme Outcomes and Course Outcomes

Appendix B

Vision: To become a Centre of Excellence in Computer Science & Engineering, moulding professionals catering to the research and professional needs of national and international organizations.

Mission: To inspire and nurture students, with up-to-date knowledge in Computer Science & Engineering, Ethics, Team Spirit, Leadership Abilities, Innovation and Creativity to come out with solutions meeting the societal needs.

Program Outcomes:

PO1: Engineering knowledge: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.

PO2: Problem analysis: Identify, formulate, research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.

PO3: Design/development of solutions: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.

PO4: Conduct investigations of complex problems: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.

PO5: Modern tool usage: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modelling to complex engineering activities with an understanding of the limitations.

PO6: The engineer and society: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.

PO7: Environment and sustainability: Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.

PO8: Ethics: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice

PO9: Individual and team work: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings

PO10: Communication: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.

PO11: Project management and finance: Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.

PO12: Life-long learning: Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

Program Specific Outcomes:

PSO1: Computer Science Specific Skills: The ability to identify, analyze and design solutions for complex engineering problems in multidisciplinary areas by understanding the core principles and concepts of computer science and thereby engage in national grand challenges.

PSO2: Programming and Software Development Skills: The ability to acquire programming efficiency by designing algorithms and applying standard practices in software project development to deliver quality software products meeting the demands of the industry.

PSO3: Professional Skills: The ability to apply the fundamentals of computer science in competitive research and to develop innovative products to meet the societal needs thereby evolving as an eminent researcher and entrepreneur.

Course Outcomes

CO1: Model and solve real world problems by applying knowledge across domains.

CO2: Develop products, processes, or technologies for sustainable and socially relevant applications.

CO3: Function effectively as an individual and as a leader in diverse teams and to comprehend and execute designated tasks.

CO4: Plan and execute tasks utilizing available resources within timelines, following ethical and professional norms.

CO5: Identify technology/research gaps and propose innovative/creative solutions.

CO6: Organize and communicate technical and scientific findings effectively in written and oral forms.

Appendix C: CO-PO-PSO Mapping

<u>Appendix C</u>

CO-PO AND CO-PSO MAPPING

	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12	PSO 1	PSO 2	PSO 3
CO 1	2	2	2	1	2	2	2	1	1	1	1	2	3		
CO 2	2	2	2		1	3	3	1	1		1	1		2	
CO 3									3	2	2	1			3
CO 4					2			3	2	2	3	2			3
CO 5	2	3	3	1	2							1	3		
CO 6					2			2	2	3	1	1			3

3/2/1: high/medium/low