

# **COGNIPLUS: AN LLM-POWERED SEMANTIC DOC EXPLORER**

Project Submitted to the  
SRM University AP, Andhra Pradesh  
for the partial fulfillment of the requirements to award the degree of

**Bachelor of Technology**  
**in**  
**Computer Science & Engineering**  
**School of Engineering & Sciences**

submitted by

**Raghu Sai K(AP20110010311)**  
**Geetha Siva Srinivas G(AP201100100316)**  
**Sai Sri Latha K(AP20110010713)**  
**Tarun Vardhan K(AP20110010297)**

Under the Guidance of  
**Assoc Prof. Dr. Krishna Prasad**



**Department of Computer Science & Engineering**  
SRM University-AP  
Neerukonda, Mangalgiri, Guntur  
Andhra Pradesh - 522 240  
May 2024

## DECLARATION

We undersigned hereby declare that the project report **CogniPlus: An LLM-Powered Semantic Doc Explorer** submitted for partial fulfillment of the requirements for the award of degree of Bachelor of Technology in the Computer Science & Engineering, SRM University-AP, is a bonafide work done by us under supervision of Assoc Prof. Dr. Krishna Prasad. This submission represents our ideas in our own words and where ideas or words of others have been included, We have adequately and accurately cited and referenced the original sources. We have also declared that We have adhered to ethics of academic honesty and integrity and have not misrepresented or fabricated any data or idea or fact or source in our submission. We understand that any violation of the above will be a cause for disciplinary action by the institute and/or the University and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been obtained. This report has not been previously formed the basis for the award of any degree from any other University.

Place : Amaravati

Date : May 8, 2024

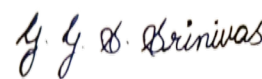
Name of student : Raghu Sai K

Signature :



Name of student : Geetha Siva Srinivas G

Signature :



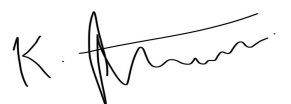
Name of student : Sai Sri Latha K

Signature :



Name of student : Tarun Vardhan K

Signature :



DEPARTMENT OF COMPUTER SCIENCE &  
ENGINEERING  
SRM University-AP  
Neerukonda, Mangalgiri, Guntur  
Andhra Pradesh - 522 240



CERTIFICATE

This is to certify that the report entitled **CogniPlus: An LLM-Powered Semantic Doc Explorer** submitted by **Raghu Sai K, Geetha Siva Srinivas G, Sai Sri Latha K, Tarun Vardhan K** to the SRM University-AP in partial fulfillment of the requirements for the award of the Degree of Master of Technology in in is a bonafide record of the project work carried out under my/our guidance and supervision. This report in any form has not been submitted to any other University or Institute for any purpose.

Project Guide

Head of Department

Name : Assoc Prof. Dr. Krishna  
Prasad

Name : Prof. Dr. Niraj Upadhayaya

Signature:

*P. Krishna Prasad*

Signature: .....

## ACKNOWLEDGMENT

We wish to record our indebtedness and thankfulness to all who helped us prepare this Project Report titled **CogniPlus: An LLM-Powered Semantic Doc Explorer** and present it satisfactorily.

We are especially thankful for our guide and supervisor Assoc Prof. Dr. Krishna Prasad in the Department of Computer Science & Engineering for giving me valuable suggestions and critical inputs in the preparation of this report. We are also thankful to Prof. Dr. Niraj Upadhayaya, Head of Department of Computer Science & Engineering for encouragement.

Our friends in our class have always been helpful and we are grateful to them for patiently listening to our presentations on our work related to the Project.

Raghu Sai K, Geetha Siva Srinivas G, Sai Sri Latha K, Tarun Vardhan K  
(Reg. No. AP20110010311, AP201100100316, AP20110010713,  
AP20110010297)

B. Tech.

Department of Computer Science & Engineering  
SRM University-AP

## ABSTRACT

'CogniPulse - Doc Explorer', an advanced document processing technology designed to simplify manuscript analysis. Authorized users can input PDFs, which are then systematically separated and raw text extracted, embedded using sophisticated techniques, and stored in a computer-generated database. The system's key feature is its semantic search engine, enabling users to query the database effectively. The algorithm ranks relevant results for display, offering a streamlined search experience. Gemini-Pro, a powerful Large Language Model (LLM), enhances system intelligence by manipulating and interpreting search results dynamically.

The user interface, powered by Flask, provides a seamless end-user experience for document analysis and knowledge retrieval. It manages conversation states, storage, and document upload sessions, integrating LLM capabilities, content retrieval, semantic search, and PDF processing. The UI design prioritizes past conversations and dynamically presents query results, enhancing user-friendliness. CogniPulse Explorer is a proficient, user-friendly, and technically robust solution for document analysis and knowledge retrieval. With its comprehensive approach, it offers an intuitive software solution to customers, making document analysis more efficient and effective.

# CONTENTS

<b>ACKNOWLEDGMENT</b>	<b>i</b>
<b>ABSTRACT</b>	<b>ii</b>
<b>LIST OF TABLES</b>	<b>iv</b>
<b>LIST OF FIGURES</b>	<b>v</b>
<b>Chapter 1. INTRODUCTION</b>	<b>1</b>
<b>Chapter 2. MOTIVATION</b>	<b>2</b>
2.1 Harnessing LLM for querying PDFs . . . . .	2
<b>Chapter 3. LITERATURE SURVEY</b>	<b>3</b>
<b>Chapter 4. METHODOLOGY AND IMPLEMENTATION</b>	<b>5</b>
4.1 NLP Preprocessing Techniques . . . . .	6
4.2 Embeddings Generation . . . . .	7
4.3 Vector Database - ChromaDB . . . . .	8
4.4 Generative AI: Large Language Models . . . . .	10
4.4.1 Semantic Search . . . . .	11
<b>Chapter 5. PDF PARSING AND ANALYSIS</b>	<b>12</b>
5.1 Text Extraction . . . . .	12
5.2 Stemming . . . . .	13
5.3 Encoding . . . . .	14
5.4 Text Cleaning . . . . .	14
5.5 Lemmatization . . . . .	15

<b>Chapter 6. CHATBOT ARCHITECTURE</b>	<b>16</b>
6.1 NLP Pipeline and Preprocessing . . . . .	17
6.2 Implementation of Embedding models . . . . .	18
6.2.1 Open AI's Embedding Models . . . . .	20
6.3 Utilizing Lang-chain LLMs . . . . .	21
6.3.1 OpenAI's and Google's Language Models	22
6.3.2 Implementation of Mistral-7B Transformers	22
<b>Chapter 7. RESULTS &amp; DISCUSSION</b>	<b>24</b>
7.1 EDA Results . . . . .	24
7.2 Stimulation Results and Analysis . . . . .	26
7.3 Discussions . . . . .	26
<b>Chapter 8. APPLICATION DESIGN</b>	<b>28</b>
8.1 Chatbot UI . . . . .	28
8.2 Chat Interface . . . . .	29
<b>Chapter 9. CONCLUSION</b>	<b>31</b>
9.1 Scope of further work . . . . .	32

## LIST OF TABLES

6.1	Open AI Embedding Models Comparison. . . . .	20
-----	--	----



## LIST OF FIGURES

4.1	Methodology . . . . .	6
4.2	Text Manipulation flowchart . . . . .	7
4.3	3D Representation of Embeddings . . . . .	8
4.4	Storing Embeddings . . . . .	9
5.1	Preprocessing . . . . .	13
6.1	Architecture . . . . .	17
6.2	Sentence Similarity [All-MiniLM-L6-v2] . . . . .	19
6.3	Sentence Similarity - II [E5-small-v2] . . . . .	19
7.1	Word Cloud . . . . .	25
7.2	Frequency Vs Word . . . . .	25
7.3	Stimulation results of Mistral-7B . . . . .	26
8.1	Application Architecture . . . . .	28
8.2	Homepage UI . . . . .	29
8.3	Chat Interface . . . . .	30

# Chapter 1

## INTRODUCTION

Comprehension of the quintessence of brilliance and initiating assuming a contraption personifies it raises powerful queries for scholars. It is broadly accepted that genuine understanding furnishes logic abilities, empowers us to examine theories, and primes for forthcoming contingencies [17]. In specific, computer intelligence investigators emphasize the advancement of technological intellect, as contradicted to naturally derived intelligence [21]. Appropriate quantification assists in perceiving acumen. To demonstrate, assessments for common in homo sapiens folks frequently embrace intelligence quotient trails [29].

In recent times, large language models (LLM) have provoked ample fascination throughout the duo of educational and manufacturing realms [2]. AI language systems hold proficiencies in dealing with multifarious assignments, dissimilar with previous models restricted to solving explicit projects [6]. Text generators are progressively utilized by personnel with crucial data necessities, for example, learners or medical cases. The examination is of preeminent to the achievements of LLMs due to numerous rationales. Linguistic models are developing and expanding with additional emerging talents, prevailing appraisal methodologies may not be adequate to scrutinize their capacities and likelihood perils [6].

## **Chapter 2**

### **MOTIVATION**

Users find it difficult to efficiently discover particular information in traditional PDF documents since they are frequently static and linear in design. Searching for keywords or scrolling through large pages manually might be annoying and time-consuming. Users may have trouble browsing long language, complicated structures, or massive datasets within PDF documents. This can result in decreased productivity, higher cognitive load, and dissatisfaction.

These issues can be resolved by the LLM-powered PDF chatbot, which gives users a conversational interface. Users may rapidly access pertinent information, ask questions, or complete activities inside the document by interacting with the chatbot in natural language.

#### **2.1 HARNESSING LLM FOR QUERYING PDFS**

LLM (Language Model) technology in PDF documents improves document comprehension capabilities. LLM models are exceptionally efficient in understanding and producing text that is like that of a human, which makes them suitable for parsing and interpreting material found in PDF files. Users have access to dynamic querying capabilities when LLM-powered chatbots are integrated into PDF documents. These chatbots can interpret natural language inquiries and respond appropriately based on the PDF's content, so improving users' querying experiences.

## Chapter 3

### LITERATURE SURVEY

This study [16] analyzes chatbots and AI methods for question-answering, focusing on the LangChain framework and LLMs for automating information retrieval. It emphasizes cosine similarity and Pinecone for vector storage. The technique's architecture includes document loading, splitting, retrieval, production, storage, and response. The study highlights the chatbot's effectiveness in handling PDF document inquiries, calling for further research to improve performance [16].

The author explains LangChain architecture in detail, focusing on its components and potential for accelerating LLM applications [26]. LangChain is highlighted for its adaptability and integration capabilities, positioning it as a key tool in the AI ecosystem. The author offers a comprehensive overview of the LangChain architecture and its implications for the future of AI technology, making it a valuable tool for academics and developers interested in utilizing LLMs for AI application development [26].

The author discusses the limits of rule-based chatbots, introduces advanced language models for text generation and support, uses Pinecone for efficient vector storage, integrates React JS for user interfaces, mentions research on conversational AI, and highlights PDF chatbots for productivity and satisfaction [23]. Users and chatbots interact more naturally due to these technologies. Language models and vector storage help chatbots respond effectively to user queries, leading to tailored discussions.

React JS interfaces enhance user experiences by personalizing chatbot

interactions. Conversational AI can improve customer satisfaction and business operations. PDF chatbots further enhance efficiency and satisfaction with prompt responses.

This article[22] presents an in-depth analysis of the current research and advancements in AI-driven PDF document processing, specifically focusing on vehicle manuals. It showcases the significant progress in deep learning and natural language processing, highlighting the role of generative AI and Large Language Models (LLMs) in revolutionizing various applications. The research delves into the shift from manual creation to machine-generated personality tests, emphasizing cutting-edge natural language processing techniques [22]. Additionally, it explores the challenges and opportunities in the automotive industry, particularly with the integration of intelligent IoT for Automobile Industry 4.0.

## Chapter 4

### METHODOLOGY AND IMPLEMENTATION

Computerization of schooling has ushered in an immense volume of internet-based resources that are feasibly beneficial for polyglots to rehearse their literacy abilities [24]. Being absorbed in a chatty context is essential at the time of acquiring knowledge to articulate a fresh dialect [24]. Individual software for facilitating subject capability is a conversational bot (chatbot), a configuration of machine intelligence program [9]. Chatbot is specified as a system-implemented tool that replicates the individual clients interacting with it [11]. These are forms of machine intelligence where the system software gives the answers to queries asked by the users depending on its extant wisdom. Usually, virtual assistants are intentionally arranged on online platforms that interact with clients like the main page or reach out page.

In computational linguistics and automated learning, the voyage from the preliminary process to using Large Language Models (LLMs) consists of a sequence of vital measures that mold the efficacy and reliability of document-oriented applications. The introductory stage, data wrangling (preprocessing), establishes the groundwork by refining and sorting unprocessed test data. Preprocessing ensures the data is regularized and primed for the following process. After initial processing, the script is commonly altered to numeric value-based illustrations familiar as embeddings.

Embeddings grasp meaningful and situational details about terminology or idioms in uninterrupted vector space. Used Word2Vec to generate

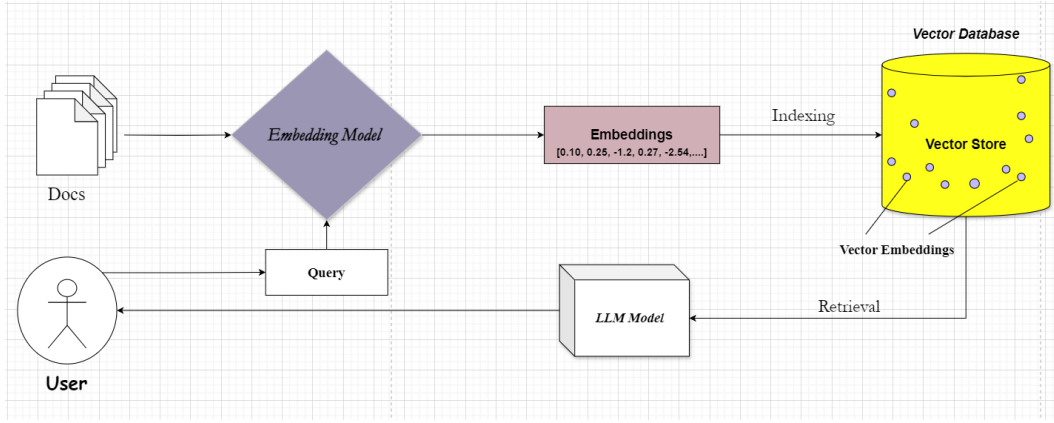


Figure 4.1: Methodology

purposeful interpretations that maintain connections among terms. These encodings cater as entries for varied computational intelligence frameworks, empowering them to comprehend and operate with literary information more efficiently. Formerly the integrations are created, and they can be archived in a vector catalog for streamlined recovery and implementation in follow-up activities like opinion mining, document categorization, and linguistic creation propelled by big-scale language processors like Generative Pre-trained Transformer 3. These prototypes exploit the opulent contextual knowledge enciphered in embeddings to create anthropomorphic scripts and assist sophisticated text analysis applications over different sectors.

## 4.1 NLP PREPROCESSING TECHNIQUES

Preprocessing is commonly the primary phase in the process flow of an NLP, with a likelihood influence on its ultimate execution [3]. Information extraction is applied for the discovery of valuable data from a massive quantity of information. Terms are regularly regarded as the fundamental components of writings for abundant communication systems, inclusive of Anglo-Saxon [4]. Information Retrieval methods are leveraged to put

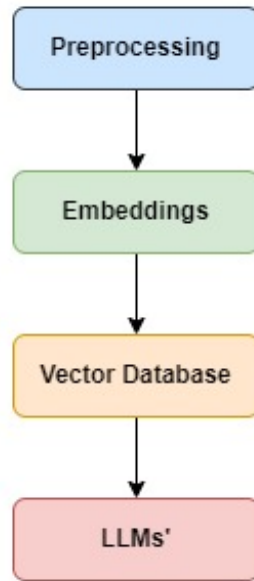


Figure 4.2: Text Manipulation flowchart

into practice and tackle unique classifications of investigation hurdles [28]. The foremost component in the text analysis conduit is a text segmenter (tokenizer) which alters literature to series involving vocabulary [3]. Nevertheless, in practice, additional preliminary processing tactics can be furthermore practiced collaboratively with lexical analysis [3]. These consist of stemming, recapitalization, and phrase categorization, among others. Text analytics are applied in diverse categories of study areas including computational linguistics, search retrieval, labeling, and cluster analysis [28].

## 4.2 EMBEDDINGS GENERATION

Encodings have been one of the predominant crucial areas in computational linguistics for the previous ten years, text analysis has upgraded enormously after the triumph of word vectorization methods like word embedding model [5] [8]. Text representations have developed into an indispensable portion of present-day language understanding platforms,



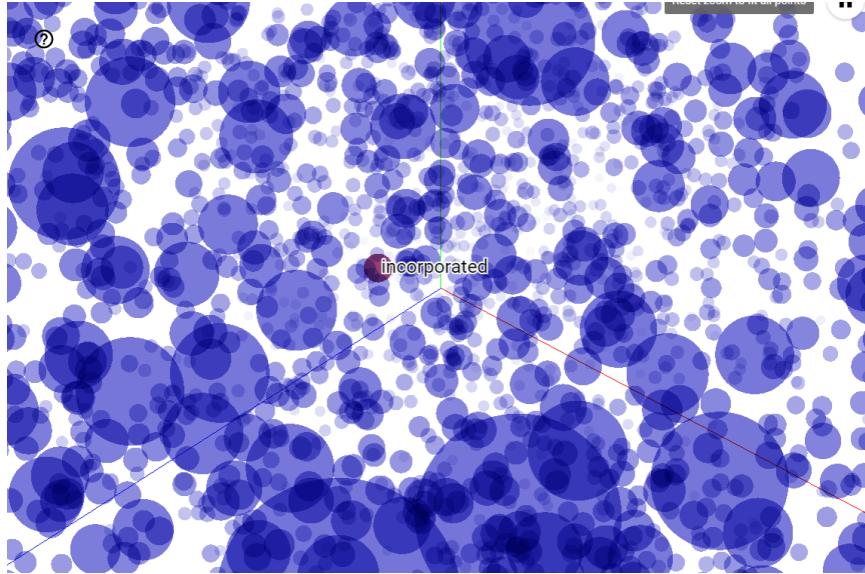


Figure 4.3: 3D Representation of Embeddings

particularly high-level neural network architectures [7]. Word embedding is an attribute understanding method whose objectives are to translate lexicon from a word stock into dimensions of rational and irrational numbers in a reduced-dimensional space, applied as the inherent entry portrayal, be an excellent strength for an extensive advantage for an extensive diversity of computational linguistics chore [19].

The above figure represents how the words are converted into embeddings that are represented in a 3D plane

### 4.3 VECTOR DATABASE - CHROMADB

Embeddings are just converted into an array of numbers known as vectors. Each vector has a finite count of proportions, which can be scoped from myriad, contingent on the intricacy and fineness of the materials. Vector contains patterns of relationships, the combination of these relationship numbers that act as multidimensional maps to measure the similarity. Once the embeddings are created, databases are also created.

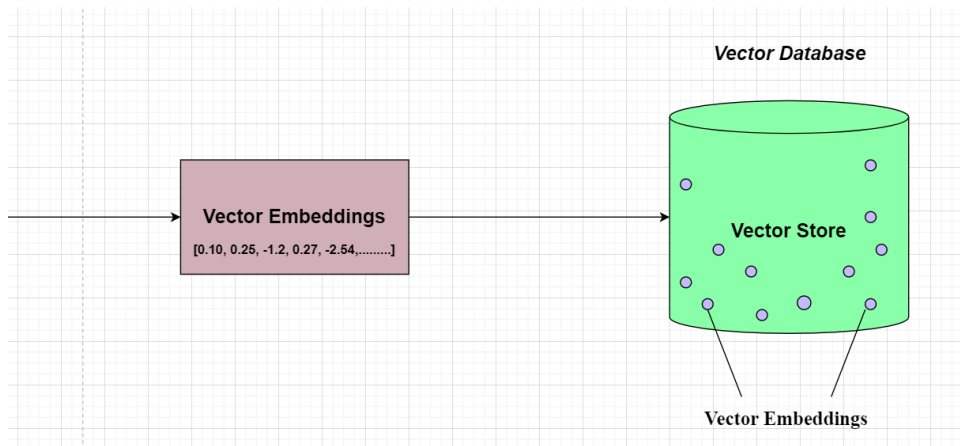


Figure 4.4: Storing Embeddings

A vector database is a kind of data warehouse that retains records as multi-dimensional scalars, which are numerical renderings of elements, worn to stockpile manifold information that is prohibited from being distinguished by a conventional database management system [14].

Most widely used and open-source databases, such as Pinecone, ChromaDB, FIASS, and MongoDB, allow users to build an index or collection in which to keep embeddings and the relevant information from which they were derived. Users just need to authenticate the database with their API keys to use it as a retriever connecting to LLM's QA interface.

We can use this in several ways as follows

- Searching - Outcomes are ordered by pertinence to the inquiry array
- Clustering - Scripts chords are bundled by analogy
- Recommendations - Objects with correlated written material threads are advised
- Classification - Text strings are classified by their most similar labels.

### **Advantages**

- Faster Research and Optimally stores the data.

## 4.4 GENERATIVE AI: LARGE LANGUAGE MODELS

LLMs are a valuable bound onward in the field of automated intelligence and language processing. Models like GPT-3.5 and GPT-4 are eminent by their gigantic scale holding thousands of millions of factors tweaked through wide-ranging training on huge datasets collected from the world wide web. This exhausted learning furnishes natural language processing models with the capability to understand and develop literature that intimately imitates verbal communication, creating extremely adaptable devices with varied functions spanning trades.

Large Language Models (LLMs) come in all shapes and sizes, and each one has its unique talents and quirks. One of the main ways they differ is in their size and the number of parameters they have. This number of parameters reflects the model's complexity, and it also affects how much computer power it needs to run. Larger models, with more parameters, are often too complex for regular computers to handle, so they need special hardware called Graphics Processing Units (GPUs) to do certain tasks. Fortunately, researchers have been making significant strides in optimizing LLM architectures and developing techniques like quantization to enable efficient execution on CPUs. Quantization refers to the process of reducing the precision of the model's internal calculations, which can dramatically reduce the computational resources required for inference without sacrificing significant accuracy.

Individuals determining the strong points of AI models is their potential to carry out a range of communication-based tasks with impressive exactness and consistency. The depth of these LLMs' training to record detailed language-related shades, circumstances, and feelings, leads to outcomes that are accordingly pertinent and language-savvy. Apart from this,

Large Language Models also face obstacles such as prejudices in the learning process, dilemmas encircling their implementation, and the requirement for rigorous analysis models to guarantee the dependability and superiority of their outcomes.

#### **4.4.1 Semantic Search**

Operations which include internet services and metadata-driven web are functioning to develop an online realm of dispersed AI systems interpretable information [13]. Semantic web, elongation of the present-day internet in which data is specified, clear-cut implication, more advantageous empowering systems, and populace to collaborate in alliance [13]. Linked Data offers several innovations to upgrade mankind and system partnership in cyberspace [18].

In recent times various types of semantic search techniques have been disclosed. Semantic search is not searching using exact keyword matching but understanding the intent of the user query and using the context to perform the search. Their scope of use and their implementation are diverse. Generally well-liked internet search platforms for instance Google, Ask and AltaVista, the leading results on seek for 'seat' are about SEAT, the Spanish car maker [27]. It is the challenge similar to keyword-centric probing that is compelling the expedition for semantic search engines. Similar to the World Wide Web, the progress of the Web of Data will be propelled by solutions that employ it [20]. It is a program of the linked data to explore. Semantic search is a report-fetching workflow that leverages field expertise [13] [20]. The objective of meaning-based search is to develop orthodox exploration techniques by harnessing context-based descriptive data [27].

## **Chapter 5**

### **PDF PARSING AND ANALYSIS**

Machine software produces a massive amount of information day to day in their records, automatically logged folders or additional reviews [15]. Portable Document Format (PDF) has turned into an unofficial benchmark for swapping digital manuscripts, for graphic depiction as excellently as for outputting [10]. This also evolved a widespread shipment for malicious software, and the former task has emphasized attributes that steer to safeguarding dilemmas [10]. Phases involved in PDF parsing and analysis are explained below.

#### **5.1 TEXT EXTRACTION**

Data extraction from a digital document includes the extraction of text-based content for instance main text, headlines, section titles, annotations, and descriptive information such as creators' identities, date of publication, and title information. The following starting step is essential as it sets the basis for following document handling tasks. Retrieving this knowledge precisely is crucial for implementations spanning from file categorization and exploration to linguistic analysis undertakings similar to opinion assessment, content retrieval, and brief synthesis. Efficient written material retrieval techniques guarantee that pertinent information and data descriptors are documented thoroughly and truthfully, facilitating subsequent and understanding production.

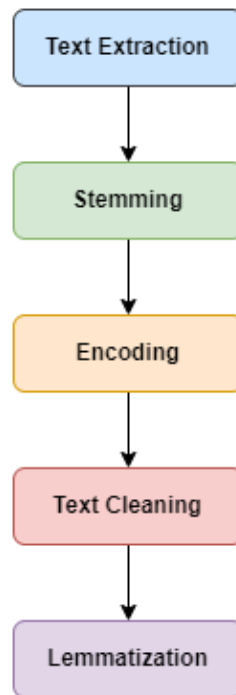


Figure 5.1: Preprocessing

## 5.2 STEMMING

Stemming is a lexical procedure employed in NLP to minimize terms to their core or root configuration, acknowledged as the stem. By eliminating endings and beginnings, stemming targets streamline the terms and classify modifications of the identical term collectively. Taking example terms such as “dancing”, “dances” and “danced” are all stemmed from the term “dance”. The aforementioned method aids in lessening the terminology dimension by integrating alike terms, as a result enhancing formatting and improving the proficiency of document evaluation algorithms. Nevertheless, stemming may sporadically result in hyper-stemming or subpar stemming, wherein the stem is abundantly clarified or erroneous, respectively.

### 5.3 ENCODING

Encoding in NLP entails converting language data into a digital format applicable to automated learning techniques. A technique named One-hot transformation indicates every term as machine code, with a 1 at the lexeme's position and 0s in other places, making simpler nominal dataset managing. Distributed word representations such as Continuous Bag of Words (CBOW), Word2Vec or Global Vectors for Word Representations (GLoVe) encode lexical associations amid linguistic units by converting them to high-dimensional vector renderings in an endless space, assisting superior setting perception. Encrypting tactics play a critical part in joining the rift between unprocessed written material and mathematical frameworks, enabling NLP users with resilient logical skills.

### 5.4 TEXT CLEANING

Text cleaning is one of the important stages in the preprocessing phase of linguistic analysis, it dreams to improve the caliber and uniformity of linguistic records. This phase includes various important actions. Primarily, this involves eliminating disturbance and unnecessary components like non-alphabetic characters, orthographic symbols, and special characters. It aids in formatting the text and making it simpler for consequent assessment. Next, text preprocessing frequently contains small letters of whole content to confirm individuality in presentation, assuring that differences in capitalization ("LETTER" or "letter") do not impact the analysis. Furthermore, well-known terms such as terminated words (for example: "and", and "is") are rarely removed during the cleaning process as they do not contribute meaning to the analysis and can cut results. Finally, data cleansing encom-

passes managing shortenings and acronyms by elaborating them to their expanded versions, securing that the content is fixed and set for subsequent handling in NLP tasks such as sentiment detection, theme extraction, and content labeling.

## 5.5 LEMMATIZATION

Lemmatization is a language-related procedure that moves further than straightforward forward stemming by contemplating the circumstances and syntactic errors of terms. Dissimilar to stemming, which roughly removes postfixes to decrease terms to their root words, morphological analysis faithfully pinpoints the canonical form alternatively base form depending on its grammatical category and its adjacent setting in a statement. By taking examples like “creating” is the verbal form and “created” is the past tense, the two words are canonicalized to “create”, guaranteeing linguistic accuracy and maintaining contextual understanding. The present exactness in term standardization adds substantially to linguistic analysis tasks like knowledge extraction, mood assessment, and translation automation by advancing the standard and dependability of computational linguistic methods.



## Chapter 6

# CHATBOT ARCHITECTURE

As the innovation carries on improvement, connecting the disparity in the middle of methodical information establishment and functional utilization turns into an imperative issue. Although the web of linked data possesses enormous capability for changing information handling and facts spreading, its complete accomplishment depends on the easy-to-use requirements of multifaceted clients. Endeavors to improve the user-friendliness and inclusively of context-aware digital tools are important in promoting extensive acceptance and unleashing the revolutionary potency of systematic research-based expertise for numerous sectors and parties involved. Dealing with the interface mentioned above obstacles is important for speeding up the assimilation of ontology-based technologies into commonplace tasks and determination procedures, eventually propelling novelty and enabling uninterrupted teamwork in the technical network.

Systematizing research-based information in methodical techniques turns into gradually crucial [1]. Semantic internet-based applications initiate not as fast as anticipated, minimum about real-world programs and customers [12]. These days this never transpired still and as a few findings draw attention to [25], this is because of the section to the reality that clients uncover it extremely problematic to utilize. A component of the primary factor for this is the absence of implementation the predominant context-based internet user experiences are nonetheless underdeveloped from the ease of use and reachability stances [12]. In the direction of surmounting

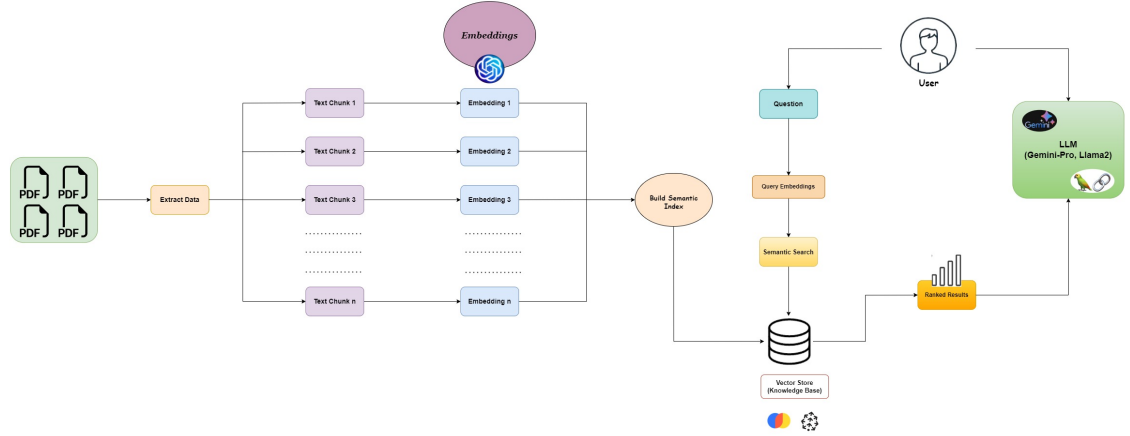


Figure 6.1: Architecture

the previously mentioned concerns we put forward our framework termed as “**CogniPlus**”. The setup and functioning of the suggested automated messaging system are mentioned above.

## 6.1 NLP PIPELINE AND PREPROCESSING

The primary phase of our chatbot is portable document format parsing and analysis. In this process extraction and interpretation are the two phases that are performed on the data of the PDF documents. Parsing phase, here the data is broken in a structured manner for analysis and also helps in information retrieval and keyword extraction. Extraction of data implementation takes by using various libraries and tools that are available in various computing languages. For our work, we used the library termed PyPDF2 and also PDFReader from langchain which is a Python library. It aids in extracting the text and image. Also provides the functions and techniques to navigate through the PDF framework and restore the required information. The data extracted from the provided document further also perform various analyses like text cleaning, encoding, and lemmatization.

This methodology is important for document management, controlling the efficient data extraction and decision-making from PDF files.

The next phase is the embedding stage. In the context of machine learning and textual linguistics, embeddings are the words or phrases that are in the dense vector representations in continuous vector space. For our work, we used the embedding architecture termed sentence transformers. In this transformer, we used the All-MiniLM-L6-V2 and E5-small-v2 models.

## 6.2 IMPLEMENTATION OF EMBEDDING MODELS

All-MiniLM-L6-V2 refers to a specific variant of the MiniLM model architecture. It is a particular specification of the MiniLM model with six layers and embodies an extra concise and resourceful variety of language frameworks fitted for different NLP works, comprising classification of text, understanding the language, and production. The architecture targets to develop sentence embeddings on huge sentence-length datasets utilizing the autonomous comparative understanding purpose. The main intention is used as a sentence and small para encrypter. The processing text outputs a vector that catches the semantic data. It is used in daily applications like sentence similarity and information retrieval etc.

For instance, let's take the source sentence "I Love You" and the three sentences to compare to the source sentences are,

- ' I adore you '
- ' I admire you '
- ' I like you '

The sentence similarity for "I Like You" is 60.9 percent and for "I Admire You" is 63.4 percent whereas for 'I hate you' is 57 percent. The process-

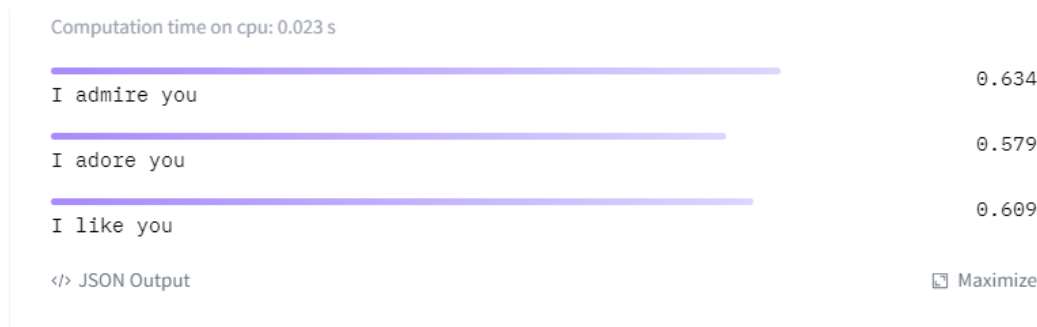


Figure 6.2: Sentence Similarity [All-MiniLM-L6-v2]

ing input text longer than 256-word pieces is cut. The model is fine-tuned by performing the cosine similarity from every individual sentence pair from the collection, implementing the cross entropy loss by juxtaposing with real pairs. It matches the sentences and paragraphs with 384-dimensional dense vector space.

"E5-small-v2" likely refers to a specific variant or version of a transformer-based model called "E5-small". This model only works for English texts. A similar example is used for E5-small-v2. The sentence similarity for "I Like You" is 93.9 percent and for "I Admire You" is 89.5 percent. As mentioned the model works for only English texts giving the higher sentence similarities. Long texts will be limited to 512 tokens.

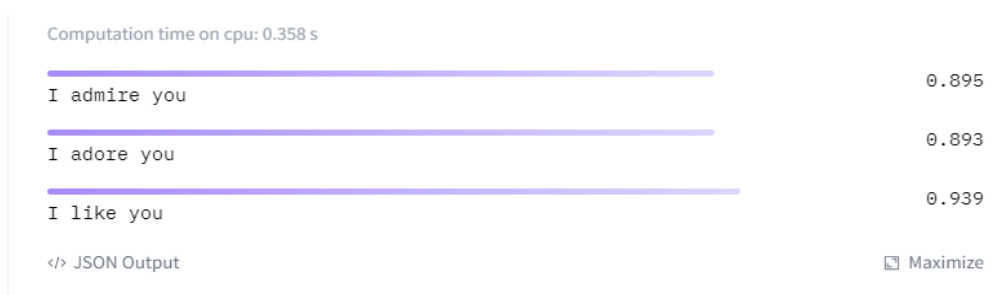


Figure 6.3: Sentence Similarity - II [E5-small-v2]

### 6.2.1 Open AI's Embedding Models

We all know that Open AI's open-source ChatGPT has grown in popularity since its launch. Furthermore, Open AI's embedded models have obtained high scores in the MTEB benchmark. In our project, we attempted to identify the differences between the various models such as text-embedding-ada-002, text-embedding-3-small, and text-embedding-3-large.

Though the architecture and data used to train these models have not been made public, it has been determined that they are largely organized similarly to transformers. The text-embedding-Ada-002 model was launched in December 2022, outranking all other embedding models on the MTEB leaderboard board, and quickly became the top embed model of the year. The other two models, 'text-embedding-3-small' and 'text-embedding-3-large', were inspired by the Ada-002 model and were constructed similarly, but with higher efficiency and low memory usage.

These models may be accessed using OpenAI's API by giving the model name in your request. The quantity of tokens processed determines the cost of utilizing these models and might vary depending on your use. OpenAI provides many price levels, with billing often occurring every month. Users may check the usage and pricing data using the OpenAI API dashboard.

<b>Models</b>	<b>text-embed-ada-002</b>	<b>text-embed-3-small</b>	<b>text-embed-3-large</b>
<b>Performances</b>	Previously Top	Moderate	Current Top
<b>Dimensions</b>	1536	1536	3072 (Adjustable)
<b>Max-Tokens</b>	2048	2048	2048 (Adjustable)
<b>BenchMark</b>	Higher	Moderate	Highest
<b>Use case</b>	Existing workflows	Cost-conscious	Top-notch accuracy
<b>Size of Model</b>	Large	Small	Largest

Table 6.1: Open AI Embedding Models Comparison.

A vector database is a kind of data store used for storing and controlling a group of information points symbolized as vectors, usually in a high-dimensional domain. These record storage systems are usually utilized in scenarios like GIS, machine learning, and bioinformatics. For our architecture purpose we used the ChromaDB, a datastore handling system especially planned for retaining and examining massive hereditary details. It is a freely available embedding information system.

Chroma allows it simply to develop a Large Language Model application by empowering information, realities, and abilities interchangeable with linguistic models. It is personalized to tackle the huge volume of genetic data produced by the latest sequencing models, offering sufficient retention and recovery systems improved for genomics data. It also allows the appliances to hold encodings and their descriptive information, implant PDFs and search inquiries and lastly explore representations. Chroma emphasizes uncomplicatedness, programmer efficiency, and examination on high exploration. It also occurs to be extremely fast.

### **6.3 UTILIZING LANG-CHAIN LLMS**

Lang-chain LLMs, which stand for Language-Chained Large Language Modeling, are a unique paradigm in the field of language modeling. Unlike traditional models, which act as independent units, Lang-chain LLMs are intended to collaborate in a chain-like approach. This linked structure allows them to combine the qualities of many models, resulting in increased performance and efficiency.

### **6.3.1 OpenAI's and Google's Language Models**

Furthermore, the integration of langchain with both OpenAI and Google gives us a substantial advantage in using the capabilities of the OpenAI GPT-3.5 Language model using OpenAI's API key, as described in OpenAI's API documentation. Similarly, to use Google's Generative AI models, such as Gemini-Pro and Gemini-Vision, users must first get an API key from Google's developer page. By setting these API keys as environmental variables, we may protect sensitive information from public access.

In our project, we employed OpenAI's powerful GPT-3.5 and Google's modern Gemini-Pro models. These models give very consistent outputs, with excellent organization, structure, and relevance to the input data. Importantly, these models operate effectively without the requirement for GPU support, giving several advantages that have established them as the leading open-source systems in use today.

### **6.3.2 Implementation of Mistral-7B Transformers**

Also on top of these models, we have used the Mistral-7B-v0.1 model which comes from the transformers package and deployed via Huggingface repo id as 'mistralai/Mistral-7B-v0.1'. The Mistral-7B models were developed by Mistral, an AI company that outperforms the Llama 2 13B (params) with only 7Billion parameters which is nearly halfway less than Llama 2. To use this model, firstly we need to grab an HF token ( Access token) and also grant access to the repo by allowing certain conditions.

Unlike GPT-3.5 and Gemini models which can run on CPU, Mistral needs a GPU to work on. That defines the strength and the complexity of the mistral model. As of now, we're working on non-GPU (CPU) machines. As loading the entire model requires GPU, instead we've imported the model

as a pipeline using tokenizer - which comes from the Autotokenizer and set the Quantization configuration.

To load our model with 4-bit precision, we will use a 4-bit quantization with an NF4-type setup and the BitsAndBytes tool. This will allow us to accelerate the model loading process while reducing memory utilization, making it possible to operate on Google Colab or desktop GPUs. We intend to use the Transformers library's pipeline function to generate a response based on the specified prompt.



## Chapter 7

### RESULTS & DISCUSSION

After testing other embedding models like all-miniLM-L6 and e5-small along with open AI's embed models, as well as language models, we determined that the Mistral-7B language model outperformed the others in terms of accuracy and relevance. It's massive, and Mistral 7b's uniqueness derives from its multi-modal generative capabilities, which allow it to create new material, connecting with the next generation of generative AI. However, the fundamental difficulty with this model is that it cannot be used as a commerce web application because of its memory utilization, as it requires a GPU to function properly.

That's when Google's New Generative AI model comes in, giving the same content and precision but using less memory. We have launched an online web application entitled '**CogniPlus - A Semantic Doc Explorer**' by employing the Gemini-Pro model.

#### 7.1 EDA RESULTS

Exploratory Data Analysis[EDA] refers to the process of visually reviewing and analyzing data to understand the underlying patterns. It also entails producing graphical representations and understanding the data used to create the chatbot. Because a substantial percentage of the data comes from PDF files including texts, words, and phrases, we performed fundamental graphing to obtain a better grasp of the document's contents.



Figure 7.1: Word Cloud

Furthermore, we have successfully generated a visual representation in the form of a bar graph that shows the frequency of the most regularly used terms in the text.

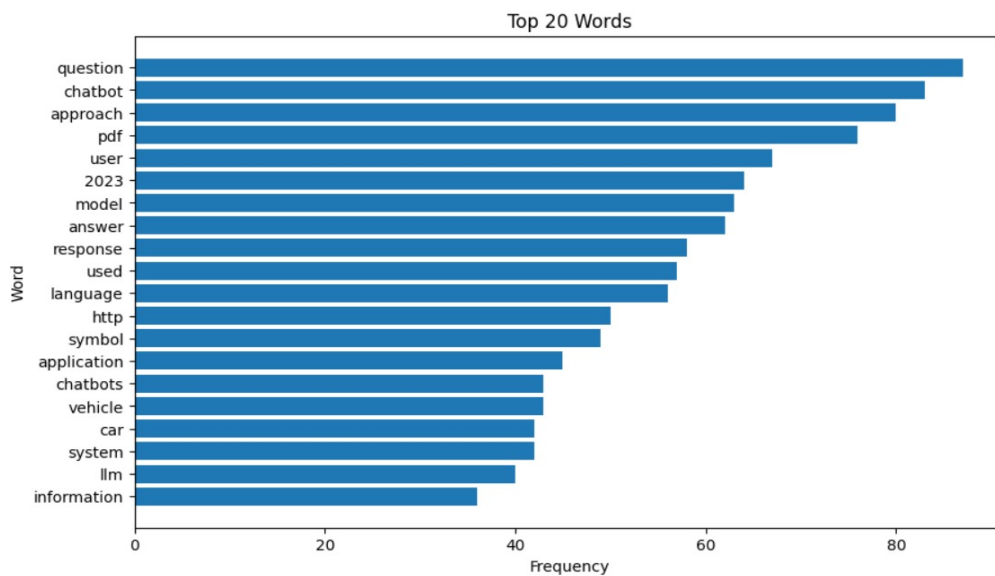


Figure 7.2: Frequency Vs Word

## 7.2 STIMULATION RESULTS AND ANALYSIS

The Mistral-7B variant required a minimum of 15GB of GPU RAM. Due to a scarcity of GPU workstations, we used Google Colab's Run type feature and configured it to use the T4-GPU (Tesla's T4, a GPU card utilizing Turing architecture particularly designed to improve deep learning model inference).

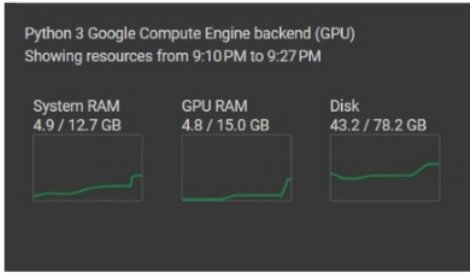


Figure 7.1: Initial GPU Consumption

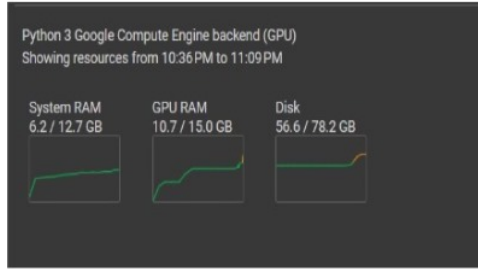


Figure 7.2: GPU consumption at loading Model

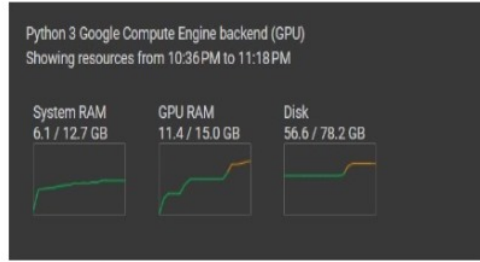


Figure 7.3: GPU consumption after getting response

Figure 7.3: Stimulation results of Mistral-7B

## 7.3 DISCUSSIONS

Implementing the PDF Chatbot utilizing Langchain, Chromadb, and the Google Gemini architecture marks a big step forward in financial chatbots. By merging modern technology, we were able to build a chatbot that engages consumers in discussions and gives individualized and accurate financial advice.

The integrated use of Langchain and Chromadb enabled the chatbot

to study and understand user data in real time, resulting in more tailored answers. Furthermore, the Google Gemini model improved the chatbot's natural language processing abilities, allowing it to grasp complicated queries and offer meaningful responses.

Overall, the PDF Chatbot illustrates the feasibility of mixing many technologies to produce a very successful and user-friendly chatbot. Future research might focus on improving the chatbot's customization features and increasing its skills to cover various financial issues.

## Chapter 8

# APPLICATION DESIGN

Using the well-known web framework Flask, we have successfully created an online web application that functions as a prototype by closely following the approach and incorporating a wide range of functionality. We have carefully tested it on a local server or host and systematically improved it through several answers and revisions.

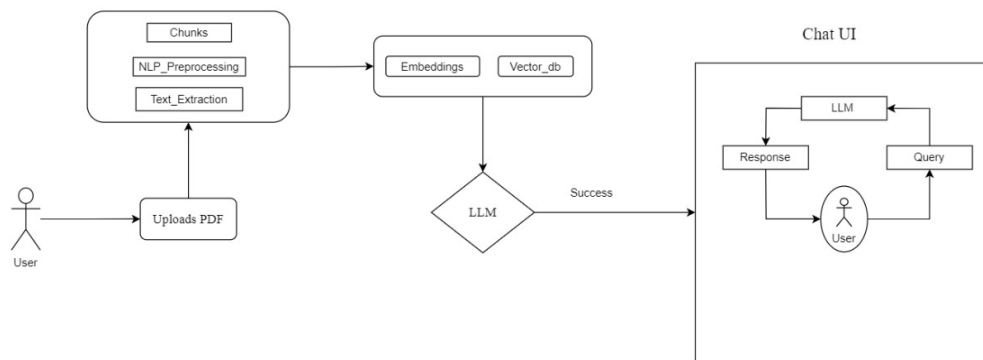


Figure 8.1: Application Architecture

### 8.1 CHATBOT UI

After extensive research and testing various chatbot interfaces available online, we have finally determined which one best meets our needs. Recognizing that the User Interface plays a major role in attracting users to websites is important. We have incorporated a well-known style frame-

work called 'Tailwindcss' and implemented 'Flask' for routing functions throughout the User Interface development process.

Our application's homepage features a drag-and-drop box for uploading PDF files, and after submitted, a loader appears. In the backend, the PDF files were processed, and text extraction was performed. Soon after, the complete text material would be divided into chunks and embeddings would be generated, which would most likely be stored in a vector database. When all of the processing is complete without issue, the user will be redirected to the chat interface.

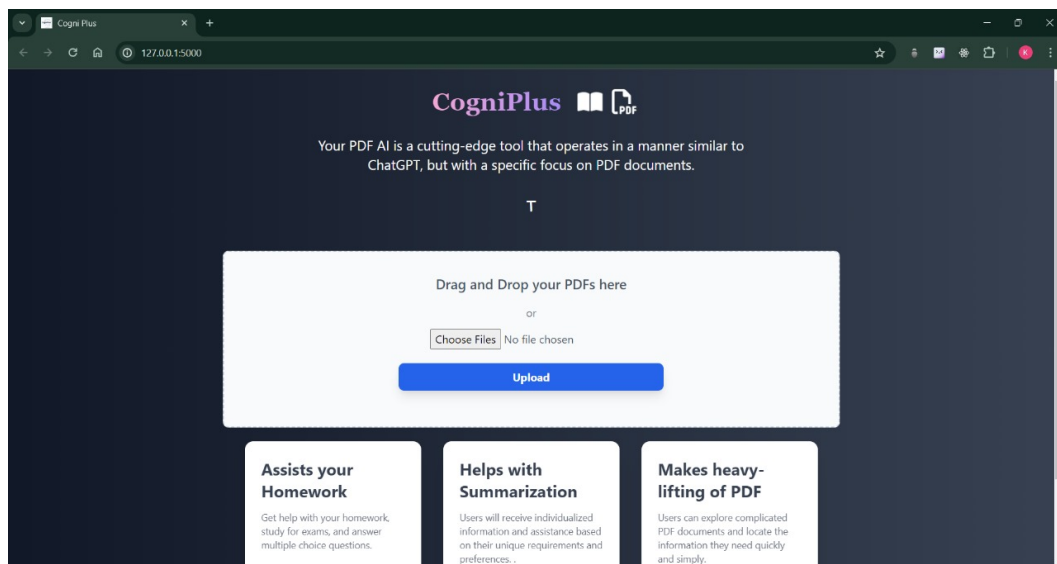


Figure 8.2: Homepage UI

## 8.2 CHAT INTERFACE

Users may quickly access their uploaded papers by scrolling to the left sidebar of the Chat Interface. Simply clicking on the names of the uploaded files will open the PDF in a new tab, offering a quick and easy method to access and examine the material.

We are now using ChatGPT-3.5 as the Language Model (LLM) to build this web-based chatbot because of its flawless post-processing capabilities in managing produced answers. In contrast, the Gemini Model requires careful management of responses and further processing. Users can enter their questions into the text section and submit them. Subsequently, our chatbot processes the query by extracting relevant information from the submitted text. The discovered relevant documents are subsequently passed to the Language Model (LLM) for additional processing and output refining.

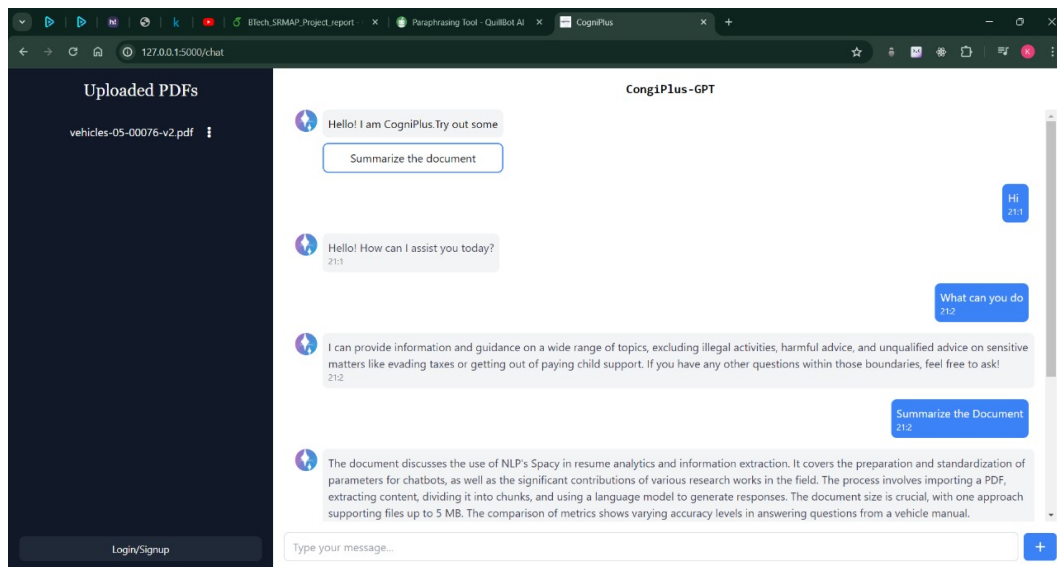


Figure 8.3: Chat Interface

## **Chapter 9**

### **CONCLUSION**

To conclude, the assimilation of leading-edge techniques like natural language preprocessing, large language models, portable document format parsing and examination structures, and chatbot framework has topped off in the advancement of a robust and adaptable stage for linguistic acquisition, paper analysis, and chatbot technology. Integrating all these systems we have designed a system that could retrieve significant understandings from the uploaded digital files, grasp the queries of the clients, and deliver accurate and captivating answers to the queries. We examined our model with Mistral - 7B arose as the superior achiever in accuracy and suitability within the embedding and LLMs. Even though its graphics processing unit storage essentiality demands some challenges for commercial web applications, the Gemini-Pro model provides similar precision and inferior memory consumption. Integrating these technologies, we developed a 'CogniPlus - A Semantic Doc Explorer'. The unification of tools like Langchain, Chromadb, and Gemini framework improves the dynamic user interaction and customized solutions to the queries. These specifications make a unique approach in the area of NLP and make the initial foundation for the upcoming interventions in language learning and AI-driven document analysis.



## 9.1 SCOPE OF FURTHER WORK

In forward-looking work, we aim to improve the modification functionality and extend the span of subjects outside the document extraction in our AI chat system. The work includes integration of sophisticated computational linguistic methods to improve the request comprehension and accuracy of replies. Moreover, we scheme to incorporate data-driven learning methods to consistently refine the virtual assistant's efficiency and resilience across the years. Investigating the assimilation of extra information sources and expanding the bot's capacity further than document content extraction to embrace a wider range of client requirements and concerns must be important for guaranteeing its sustained meaningfulness and practicality.

## BIBLIOGRAPHY

- [1] Karl Aberer, Alexey Boyarsky, Philippe Cudré-Mauroux, Gianluca Demartini, and Oleg Ruchayskiy. Sciencewise: A web-based interactive semantic platform for scientific collaboration. In *10th International Semantic Web Conference (ISWC 2011-Demo), Bonn, Germany, 2011*.
- [2] Rishi Bommasani, Drew A Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, et al. On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*, 2021.
- [3] Jose Camacho-Collados and Mohammad Taher Pilehvar. On the role of text preprocessing in neural network architectures: An evaluation study on text categorization and sentiment analysis. *arXiv preprint arXiv:1707.01780*, 2017.
- [4] Jose Camacho-Collados and Mohammad Taher Pilehvar. From word to sense embeddings: A survey on vector representations of meaning. *Journal of Artificial Intelligence Research*, 63:743–788, 2018.
- [5] Jose Camacho-Collados and Mohammad Taher Pilehvar. Embeddings in natural language processing. In *Proceedings of the 28th international conference on computational linguistics: tutorial abstracts*, pages 10–15, 2020.
- [6] Yupeng Chang, Xu Wang, Jindong Wang, Yuan Wu, Linyi Yang, Kaijie Zhu, Hao Chen, Xiaoyuan Yi, Cunxiang Wang, Yidong Wang, et al. A

- survey on evaluation of large language models. *ACM Transactions on Intelligent Systems and Technology*, 2023.
- [7] Qingyu Chen, Yifan Peng, and Zhiyong Lu. Biosentvec: creating sentence embeddings for biomedical texts. In *2019 IEEE International Conference on Healthcare Informatics (ICHI)*, pages 1–5. IEEE, 2019.
  - [8] Zimin Chen and Martin Monperrus. A literature study of embeddings on source code. *arXiv preprint arXiv:1904.03061*, 2019.
  - [9] Michelle Ehrenpreis and J DeLooper. Implementing a chatbot on a library website. *Journal of Web Librarianship*, 16(2):120–142, 2022.
  - [10] Guillaume Endignoux, Olivier Levillain, and Jean-Yves Migeon. Caradoc: A pragmatic approach to pdf parsing and validation. In *2016 IEEE Security and Privacy Workshops (SPW)*, pages 126–139. Ieee, 2016.
  - [11] Darlene Fichter and Jeff Wisniewski. Chatbots introduce conversational user interfaces. *Online Searcher*, 41(1):56–58, 2017.
  - [12] Roberto García, Juan Manuel Gimeno, Ferran Perdrix, Rosa Gil, Marta Oliva, Juan Miguel López, Afra Pascual, and Montserrat Sendín. Building a usable and accessible semantic web interaction platform. *World wide web*, 13:143–167, 2010.
  - [13] Ramanathan Guha, Rob McCool, and Eric Miller. Semantic search. In *Proceedings of the 12th international conference on World Wide Web*, pages 700–709, 2003.
  - [14] Yikun Han, Chunjiang Liu, and Pengfei Wang. A comprehensive survey on vector database: Storage and retrieval technique, challenge. *arXiv preprint arXiv:2310.11703*, 2023.

- [15] Shubham Jain, Amy De Buitelir, and Enda Fallon. A review of unstructured data analysis and parsing methods. In *2020 International Conference on Emerging Smart Computing and Informatics (ESCI)*, pages 164–169. IEEE, 2020.
- [16] Mutiara Auliya Khadija, Abdul Aziz, and Wahyu Nurharjadmo. Automating information retrieval from faculty guidelines: Designing a pdf-driven chatbot powered by openai chatgpt. In *2023 International Conference on Computer, Control, Informatics and its Applications (IC3INA)*, pages 394–399. IEEE, 2023.
- [17] Jean Khalfa. What is intelligence? 1994.
- [18] Gregk Kuck. Tim berners-lee’s semantic web. *South African Journal of information management*, 6(1), 2004.
- [19] Rémi Philippe Lebrete. Word embeddings for natural language processing. Technical report, EPFL, 2016.
- [20] Christoph Mangold. A survey and classification of semantic search approaches. *International Journal of Metadata, Semantics and Ontologies*, 2(1):23–34, 2007.
- [21] John McCarthy. From here to human-level ai. *Artificial Intelligence*, 171(18):1174–1182, 2007.
- [22] Thaís Medeiros, Morsinaldo Medeiros, Mariana Azevedo, Marianne Silva, Ivanovitch Silva, and Daniel G Costa. Analysis of language-model-powered chatbots for query resolution in pdf-based automotive manuals. *Vehicles*, 5(4):1384–1399, 2023.
- [23] Arjun Pesaru, Taranveer Singh Gill, and Archit Reddy Tangella. Ai assistant for document management using lang chain and pinecone.

- [24] Sherry Ruan, Angelica Willis, Qian Yao Xu, Glenn M Davis, Liwei Jiang, Emma Brunskill, and James A Landay. Bookbuddy: Turning digital materials into interactive foreign language lessons through a voice chatbot. In *Proceedings of the sixth (2019) ACM conference on learning@scale*, pages 1–4, 2019.
- [25] Nigel Shadbolt, Tim Berners-Lee, and Wendy Hall. The semantic web revisited. *IEEE intelligent systems*, 21(3):96–101, 2006.
- [26] Oguzhan Topsakal and Tahir Cetin Akinci. Creating large language model applications utilizing langchain: A primer on developing llm apps fast. In *International Conference on Applied Engineering and Natural Sciences*, volume 1, pages 1050–1056, 2023.
- [27] Victoria Uren, Yuanguai Lei, Vanessa Lopez, Haiming Liu, Enrico Motta, and Marina Giordanino. The usability of semantic search tools: a review. *The Knowledge Engineering Review*, 22(4):361–377, 2007.
- [28] S Vijayarani, Ms J Ilamathi, Ms Nithya, et al. Preprocessing techniques for text mining-an overview. *International Journal of Computer Science & Communication Networks*, 5(1):7–16, 2015.
- [29] Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, et al. Emergent abilities of large language models. *arXiv preprint arXiv:2206.07682*, 2022.