



UNIVERSITÀ
DEGLI STUDI
DI PADOVA



DIPARTIMENTO
DI INGEGNERIA

UNIVERSITY OF PADUA

Developing an AI Chatbot for Retrieving Information from Biological Databases

CORSO DI LAUREA IN: Information Engineering

DIPARTIMENTO DI INGEGNERIA DELL'INFORMAZIONE

RELATORE: PROF. ALEXANDER MIGUEL MONZON

LAUREANDO: DANIYAL NASIRI BAVIL

ANNO ACCADEMICO: 2023/24

DATA DI LAUREA: 19/07/2024

Abstract

The rapid expansion of biological data has posed increasing challenges for researchers in accessing and navigating relevant information efficiently. Biological databases serve as comprehensive collections of structured and organized biological information, encompassing various disciplines such as genomics, proteomics, metabolomics, and other omics areas, alongside data on biological pathways, interactions, structures, and functions. Leveraging artificial intelligence techniques, particularly natural language processing (NLP) and machine learning, the chatbot aims to streamline the process of querying biological databases. This project focuses on developing the chatbot, encompassing implementation details, design considerations, data preprocessing techniques, and algorithmic approaches. As a use case, the database DisProt will be utilized to test the chatbot's functionality. Additionally, the project explores the potential benefits, limitations, and future directions of utilizing AI chatbots in biological databases.

Acknowledgements

I would like to express my deepest gratitude to those who have supported and guided me throughout the process of completing this thesis.

I wish to express my gratitude to Prof. Alexander Miguel Monzon for his endless guidance, continuous support, and patience. His vast experience and knowledge acted as a great inspiration even as his insights and feedback proved to be essential during this work's completion. I count myself lucky to have the chance to be supervised by him.

I want to thank my dear girlfriend, Hanieh Ameri, who has been truly supportive, offering immeasurable love, encouragement, and comprehension for which I'm extremely appreciative. Even in the most difficult moments, her faith in me has sustained me; she is a pillar of strength that has never crumbled. Thank you for being my cornerstone and greatest supporter.

I also wish to thank my friends, Seyed Alireza Mousavifar and Amirhashayar Maghsoud, for their emotional backing and presence with me. The jokes, suggestions, and commendations they give me have been so constructive which made my adventure more interesting than ever before. Laughter and coming up with ideas we spent together are always my wonderful memories.

Last but not least, I would like to express my gratitude to my family for supporting me to achieve this. Whether the contributions were substantial or tiny, they all contributed towards the successful completion of my thesis.

I thank you all from the bottom of my heart.

Contents

1	Introduction	5
1.1	Background	5
1.2	Problem Statement	6
1.3	Scope of the Thesis and Objectives	7
2	Literature Review	9
2.1	Biological Databases	9
2.2	Information Retrieval Techniques	10
2.3	Artificial Intelligence in Information Retrieval	10
2.4	Existing Chatbots in Various Domains	11
3	Methodology	13
3.1	Design Considerations	13
3.2	System Architecture	13
3.3	Data Sources and Preprocessing	14
3.4	Algorithmic Approaches	14
3.5	Implementation Details	15
4	Implementation	16
4.1	General Overview and Components	16
4.1.1	User Interface (UI)	16
4.1.2	Frontend	16
4.1.3	Backend	16
4.1.4	Data Processing	17
4.1.5	Embedding Generation	17
4.1.6	Database	17
4.1.7	Interaction Flow	17
4.2	Development Environment	18
4.3	Data Handling and Storage	18
4.4	Embedding Generation	19
4.5	API Creation	20
4.6	Handling Token Limitations and Chunking Data	20
4.7	User Interface	21

5	Evaluation and Testing	24
5.1	Testing Methodology	24
5.2	Performance Metrics	24
5.3	Results and Analysis	25
5.4	Examples of Chatbot Interactions	26
5.4.1	Example 1: Summary of a Protein	26
5.4.2	Example 2: Specific Information Request	27
5.4.3	Example 3: Disordered Regions in a Protein	27
5.4.4	Example 4: Evidence Count for Structural State	27
5.5	Discussion	28
6	Conclusion	29
6.1	Advantages of AI Chatbots in Bioinformatics	29
6.2	Limitations	29
6.3	Future Directions of the current chatbot	30
6.4	Final Thoughts	30

Chapter 1

Introduction

1.1 Background

The rapid increase of biological data has created a great challenge for people who want to find and use it easily. The constant growth of different kinds of data sets in genomics, proteomics, metabolomics, and other fields means that Having such genetic compounds is more needed than ever before because these areas must be fully documented. Therefore, there should be no gaps in any of them, particularly in protein databases, which are crucial because they contain a huge amount of valuable information. (See example of Protein in Figure 1.1).

Artificial intelligence (AI) comes as a changing force within this rising area; it offers promising ways of dealing with the complexities found in biological database queries. Intelligent chatbot development has entered a new phase with the application of AI techniques, specifically natural language processing (NLP) and machine learning techniques. In the field of biological databases, this enterprise aims to focus on the abilities of AI. A comprehensive approach to this goal includes the development of a chatbot involving implementation details, design considerations, data preprocessing techniques, and algorithmic methods. Using careful experimentation and updates, the chatbot's functionality

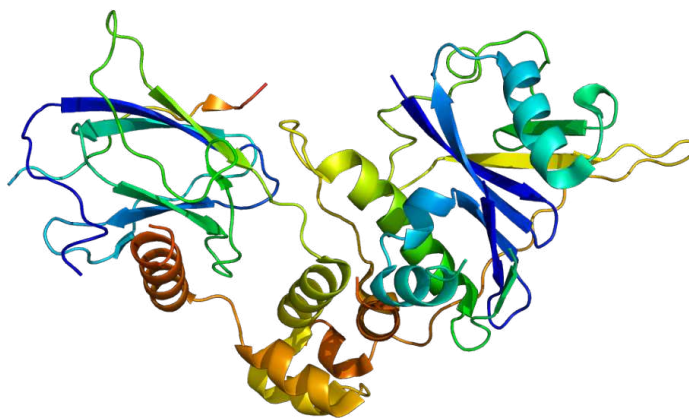


Figure 1.1: von Hippel-Lindau disease tumor suppressor - Image Credit: Wikipedia

will be thoroughly tested on the DisProt database.

In addition to focusing on the short-term objectives of chatbot development, this project is also exploring in depth the extent of the advantages that could be derived, constraints as well as the way forward of incorporating AI chatbots into biological datasets. The realization of completely new routes and approaches to increase the efficiency of studies, encourage cooperation among researchers, and improve comprehension of difficult biological datasets. This campaign is all about looking into what AI-based technology can offer.

Enabling the author to describe the completion of this study, whose aim was to investigate the most important aspect of artificial intelligence chatbots' job to enhance fast access to vast biological data, which leads to the promotion of innovative discoveries in biomedical research and data analysis field.

1.2 Problem Statement

The sudden growth in biological data generation has significantly outpaced the development of effective tools for data management and retrieval. Researchers are increasingly faced with the troubling task of accessing and navigating vast and complex datasets spanning genomics, proteomics, metabolomics, and other omics fields. The complexity of data inconsistency across several databases combined with the highly specialized nature of biological information makes these tasks very difficult since they hinder the effective use of data as well as the discovery of knowledge.

Biological databases are accessible to allow access and organization of existing info. The steepness of the learning curve that is usually encountered by users is due to the large extent of knowledge required in terms of database architectures, query languages, and terminologies that are specific to given domains. This complexity results in inefficiencies, errors, and missed chances for data analysis and interpretation. Innovation is required; it would make the dialogue between biological databases smoother: researchers could then access or use information more efficiently.

Artificial Intelligence, particularly Natural Language Processing (NLP) and Machine Learning offers a promising way to tackle these challenges. AI-powered chatbots can present user-friendly chat-based interfaces that make the querying process simpler and enable a wider range of users to access it. Nevertheless, AI chatbots merging into biological databases has still not been discovered with few studies looking at their development, putting into practice, and actual application in the present-day world.

The main goal of this thesis is to fill this void by developing an artificial intelligence chatbot that can facilitate the accessing of biological databases. The problems that this project will tackle include the following:

- 1. Complexity in Querying Biological Databases:** Researchers face difficulties in constructing and executing queries due to the complex nature of the database structures and specialized terminologies.
- 2. Fragmentation of Biological Data:** Data being spread out over several databases would make it much harder to retrieve them and also would mean that the entire biological information can not be comprehensively analyzed.
- 3. Inefficiency in Data Utilization:** Accessing and navigating biological databases can be fully time-consuming and error-prone hence reducing research productivity and efficiency.

The aim of this research is to design a well-functioning and efficient AI chatbot based on implementing details of algorithms, design considerations, data pre-processing methods, and approach to computation. To check how well the bot is responding to the user's queries, we will use the DisProt as a reference database. Finally, this thesis will try to demonstrate the advantages of AI chatbots in enhancing the accessibility and utility of biological data, while also outlining future directions for further research and development in this area.

1.3 Scope of the Thesis and Objectives

The scope of this project is to design, develop and evaluate an AI chatbot, which is specially designed for querying biological databases. This project uses Natural language processing (NLP) and machine learning techniques with an intuitive, user-friendly, effort-less interface that allows for effective data retrieval in addition to easy navigation. The project has multiple crucial steps:

- 1. Develop an AI Chatbot for Biological Databases:** The user will need a chatbot that can use biological databases for accessing information and it will specialize in DisProt. It should also possess skills to explain what he/she is looking for using normal language so the system provides correct feedback after analyzing all words meant by the request. The chatbot interprets user queries using advanced NLP and machine learning algorithms to retrieve the appropriate data from the databases. The user interaction is improved by evaluating the effectiveness of these techniques in enhancing query accuracy.
- 2. Improve Data Accessibility and Usability:** Design a chatbot used to make it easy to query biological databases to decrease the rate of user learning and improve accessibility and usage of complex biological data for the people dealing with this detailed big data.
- 3. Assess Performance and Usability:** Perform a comprehensive test of chatbot performance by checking how accurate fast friendly replies are made by the chat before getting back feedback to find out what it improves so that weaknesses can be addressed in all areas.

4. Explore Broader Implications and Future Directions: Examine biological databases with AI chatbots focusing on potential strengths and drawbacks. Further study areas and possible research should take into account the enhancement of the chatbot's competence with alternative bases and integration of other AI techniques.

With such objectives to consider, the goal is to demonstrate that AI chatbots are capable of making the process of retrieving biological data more efficient and user-friendly and consequently speeding up the monitoring of progress within biomedical research.

Chapter 2

Literature Review

This literature review explores how biological databases are linked with information retrieval (IR), artificial intelligence (AI) as well as conversational agents (chatbots). It discusses the necessity to access and interact with a wide-ranging biological data domain, which is complex, and outlines the image of chatbots involving AI that have been developed to answer as fast as possible to these issues.

2.1 Biological Databases

Modern life science research relies heavily on biological databases. These databases are maintained by scientists worldwide as a place to store information generated over time. Therefore, These involve digital collections spanning various fields, data categories as well as complexities, thus requiring comprehensive mechanisms and strategies for successful information retrieval and use.

- **Established bases in Biological Data:** Understanding the genetic blueprint of life has always been with the help of well-known databases such as; Gen bank [1] for DNA and RNA sequences, Ensembl [2] for comprehensive genome annotations, and the UCSC Genome Browser [3] used in visualization for analyzing genomic data. UniPort [4] is the key resource for protein sequences and their functions whereas PDB [5] stores experimental protein structures. Moreover, databases such as InterPro [6] divide the proteins into families as well as domains denoted by sequence signatures, hence clarifying their evolutionary connections and functional meaning. Metabolic pathways and networks that must be understood in cellular activities are well taken care of in HMDB [7] for human metabolites, MetaCyc [8] for metabolic pathways of various organisms and KEGG [9] followed by associated genomics. Reactome [10] And Biogrid [11] are databases that have mapped out the complex web of biological pathways and interactions beyond individual molecules, providing a systems-level perspective on cellular processes. Additionally, through various evidence sources, STRING [12] predicts and displays protein-protein interaction networks that reveal functional relationships between proteins.

2.2 Information Retrieval Techniques

Traditional information retrieval methods, despite laying down a firm basis for the search for data, frequently encounter problems when dealing with the complicated nature of biological information. In fields like life sciences, terms can often be synonyms or abbreviations that are related in complex ways, As a result, it is difficult to apply the simple keyword-matching procedures used by most internet browsers during basic keyword searches. [13]. Utilization of logical operators (AND, OR, NOT) in a Boolean search query can enhance search result accuracy; however, the user should possess some prior knowledge about database structures, expressions, and search syntax hence not easily accessible to many people. Vector space models query representation challenges occur because of biological data's highly interlinked complex nature. [13][14]

- **Semantic Search:** Many businesses have focused on semantic search due to its ability to increase search precision by analyzing the meaning of the words as they exist in a file system on the internet. Simple keyword matching does not cut it anymore, as this strategy involves algorithms that try to understand the purpose behind the user's use of certain terms and their importance in the relevant context. We will have more intuitive and efficient retrieval if we proceed this way. [15].
- **Knowledge Graph Embeddings:** To represent entities and relationships in a knowledge graph, techniques such as knowledge graph embeddings represent them as low-dimensional vectors in a continuous vector space. In other words, these embeddings are quite important as they ensure that the meaning between different entities is preserved (since these mathematical objects capture semantics), allowing for better data retrieval, reasoning, and analysis. They serve as a foundation for enhancing search engines recommendation systems, and other AI applications in general. [16].
- **Context-Aware Retrieval:** These can be the user's past activities, as well as location, and time, among other aspects. When context is included in it, retrieval systems give more appropriate answers based on the person to who they belong; and since this may involve anyone else's responses in between them, users find satisfaction as well as increase their effectiveness for discovering explicit content. [17].

2.3 Artificial Intelligence in Information Retrieval

- **Natural Language Processing:** AI is important as it helps us to understand and use human language in ways that are beneficial for information retrieval. Some of these important tasks are:

- **Named Entity Recognition (NER):** The pre-identified categories into which proper names are classified include names of persons, organizations, locations, times, quantities, monetary values, percentages, etc. [18].
 - **Relation Extraction:** Identifying and extracting meaningful relationships between entities in the text [19].
 - **Text Summarization:** Using a computer program to trim a text document so that the summary produced retains the most significant points of the original document. [20].
- **Machine Learning - Adaptive Learning for Enhanced Retrieval:** Machine learning techniques are important for information retrieval improvement because they allow systems to learn from data and change (Machine). The key areas for the machine learning techniques:
 - **Query Understanding:** Improving the system’s ability to understand and interpret user queries accurately [21].
 - **Relevance Ranking:** Using algorithms to sort them based on their relevance is the most appropriate way to order the fetched outcomes. [22].
 - **Recommender Systems:** Making suggestions to them may be guided by the way clients behave and specify their preferences [23].
 - **Deep Learning:** In information retrieval, several tasks have significantly improved by recent state-of-the-art advancement in deep learning models largely consisting of transformers:
 - **Transformer-based Language Models:** Artificial intelligence has made it possible for systems to comprehend natural language better and create it in a very human-like manner. [24].
 - **Predictive Modeling and Data Analysis:** Leveraging deep learning in trend prediction, result in anticipation and intricate data set analysis. [25].
 - **Explainable AI (XAI):** Explainable artificial intelligence refers to techniques and tools that allow human beings to understand the results of artificial intelligence systems. In Information Retrieval, explainable artificial intelligence is used for transparency, trust, and the ability to make artificial intelligence systems better [26].

2.4 Existing Chatbots in Various Domains

- **Chatbots as Research Assistants:** An AI is increasingly getting used as a research assistant aiding in literature searches, data retrieval, and even preliminary data analysis. They give an interactive method through which one can enquire huge databases and receive immediate answers thus hastening the research process. [27].

- **Data Exploration and Visualization with Conversational Interfaces:** For which purpose are conversational interfaces being used by chatbots? They enable people to make their queries on data in plain language which aids in generating visualizations or seeking information hidden beneath it. [28].
- **Personalized Learning with Intelligent Tutoring Systems:** Chatbots are used in artificial intelligence-based tutoring systems that provide individualized educational experiences. They adapt to the unique speed and style in which individuals study by offering customization features such as personal content, comments, and suggestions to improve learning [29].

Chapter 3

Methodology

The methodology employed in creating the AI chatbot intended for accessing biological databases is outlined in this chapter including design considerations, system architecture, sources of data used, and preprocessing steps as applied. It also elaborates on the algorithmic approaches and presents a thorough discussion of the implementation details which together enable one to understand the development trajectory of the chatbot.

3.1 Design Considerations

Several major factors influenced the design of the AI chatbot. Among these factors, the chatbot should be easy to use and have an interface that intuitively allows users of different levels of expertise to communicate with it efficiently. Hence a lot of care was given to user experience so that it could understand and accurately respond to natural language questions. In addition to that the system needed to be strong and scalable and it should be able to handle a lot of queries and bring back correct results in timeframes. It had to be ensured that data privacy as well as security is treated with importance being that the biological data might be very sensitive. Moreover, the design had to adhere to appropriate statutes for data protection as well as take in user security best practices.

3.2 System Architecture

The Chatbot was designed with a system architecture so that different components could easily integrate, thus ensuring that it operates and interacts efficiently. This architecture featured a front-end interface where users could input their questions and get answered. For interpretation purposes, this frontend utilized an NLP engine for translation of the questions into a form that could be understood by the system (Figure 3.1 , Figure 3.2). At the heart of the system was a backend whose components were the module for database integration and the algorithm engine for data retrieval and processing. This backend was tasked with the responsibilities of querying biological databases, manipulating the data, and creating relevant feedback.

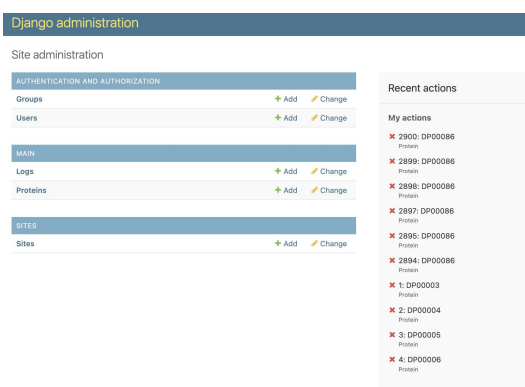


Figure 3.1: Localhost/Admin

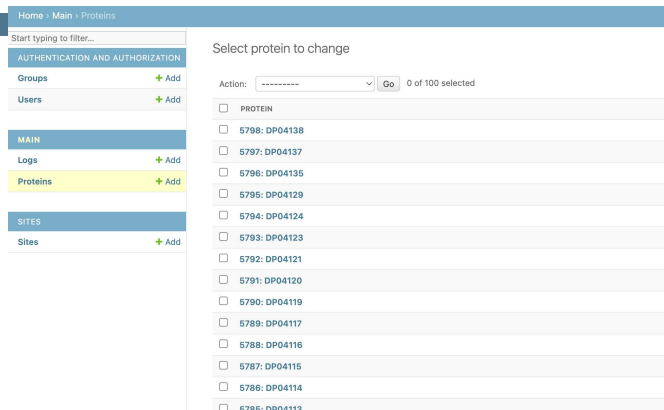


Figure 3.2: Localhost/Admin/Protein

3.3 Data Sources and Preprocessing

To ensure that the chatbot can deal with this data effectively, extensive preprocessing is essential. This ensured that the chatbot could effectively retrieve and use such data as needed while focusing on protein intrinsic disorder information, such as that found in the DisProt database (<https://disprot.org>) [Aspromonte et al. NAR].

The process included cleaning up the records by eliminating any discrepancies or mistakes, standardizing them, and creating an index to enable quick searching. Additionally, it ensured there was no repetition within the entire dataset to prevent increased redundancy when referencing text fragments, avoiding potential mismatches in syntax or grammar when viewed from different perspectives. In addition, information such as data about the document and its annotations were extracted and organized to enhance the chatbot’s interaction capabilities when handling complex questions.

3.4 Algorithmic Approaches

We used many advanced algorithms to boost our chatbot’s performance. For better user query comprehension and biological data relevance, we had to employ natural language processing methods — such as named entity recognition (NER), relation extraction, and text summarization. We used machine learning models to improve understanding and ranking relevance in queries while ensuring that the most important data was presented to users. Additionally, we applied deep learning methods, especially using Transformer-based language models to further improve the chatbot’s ability to accurately produce natural language output as well as understand inputs such as queries or conversation replies.

- **Named Entity Recognition (NER):** The use of this technique helped in recognizing and subsequently classifying different entities that are found in user queries. It may include but is not limited to protein names among other biological terms; these are some of the items that were very difficult to teach the system about. This

way, the chatbot could better contextualize what users wanted it to do, thereby assisting them more effectively.

- **Relation Extraction:** We utilized connection extraction to discover the associations between agents in our knowledge base, which facilitated us in developing meaningful insights mirroring biological links underlying our datasets.
- **Text Summarization:** Algorithms utilized for text summarization help in minimizing information overload associated with large volumes by generating precise and short summaries that users can quickly read and understand better.
- **Query Understanding and Relevance Ranking:** Artificial intelligence models were trained to interpret user requests and evaluate the relevance of fetched information for ranking it, thus putting the chatbot in the best position to provide accurate and helpful responses.
- **Transformer-based Language Models:** In anchoring the chatbot's natural language comprehension and response capabilities, advanced deep learning models like Generative pre-trained transformer (GPT) were employed. Its ability to handle complicated queries and provide coherent responses was made possible through the integration of these models.

3.5 Implementation Details

Developing the Natural Language Processing (NLP) engine was the first stage, followed by integrating the DisProt into it to construct a chatbot. It must include functions for the front end through which a user may ask questions and get immediate answers while ensuring that data fetches from the back-end system are conducted speedily and processed effectively using various optimization methods to improve different performance indicators. We conducted testing to evaluate how accurate the chatbot is compared to its response time as well as the entire user experience. Feedback from these tests was used to make the system better by fixing any problems it had and making it more effective.

In short, the strategy of creating the AI chatbot was shaped by thoughtful design specifications, the development of a strong system architecture, precise processing of information, using algorithms, and specific methods of putting it into practice. The purpose of all these endeavors was to create a highly efficient tool that would enable more effective querying of biological databases, while also simplifying access and usability of complex biological data for scholars in the field and beyond.

Chapter 4

Implementation

For the implementation of AI chatbots in biological databases, several significant parts play different roles in the functionality of this system. Below is an extensive view of the elements and their interplay. This text chapter contains detailed information on the components and architecture of the Artificial Intelligent chatbot system such as interface design, front-end surfacing as well as back-end data processing, embedding creation, database design, and interaction flow.

4.1 General Overview and Components

4.1.1 User Interface (UI)

- **Components:** Chat box, input fields, and buttons.
- **Technologies:** HTML, CSS, JavaScript.
- **Functionality:** It allows users to interact with the chatbot by typing queries and getting responses. It captures user input in the UI while also displaying chatbot replies, hence changing dynamically as per user interactions.

4.1.2 Frontend

- **Technologies:** HTML, CSS, JavaScript.
- **Functionality:** The user interface is managed, the user input is captured, and queries are sent to the backend through API requests. It makes sure that the user experience is smooth and fast.

4.1.3 Backend

- **Technologies:** Django framework, Django REST Framework.
- **Components:** API endpoints, data processing modules, embedding generation.

- **Functionality:** It manages user inquiries, processes information, creates embeddings, and interacts with the database. The backend coordinates the entire process of work starting from receiving requests up to giving answers.

4.1.4 Data Processing

- **Steps:** Data cleaning, normalization, chunking.
- **Functionality:** It also requires cleaning of data to ensure that it is consistent removing inconsistencies, making sure normalization guarantees a uniform format, and dividing extensive texts into small pieces to handle token limitations imposed by the embedding system.

4.1.5 Embedding Generation

- **Technologies:** OpenAI API.
- **Functionality:** It turns the text data into multi-dimensional vectors (embeddings) that seize the semantic content of the text. Large texts are usually divided into pieces due to the limitation of tokens, then embeddings are created for every piece. Thereafter these embeddings are joined together to come up with a coherent representation of the whole text.

4.1.6 Database

- **Technologies:** PostgreSQL.
- **Components:** Protein data, embedding vectors.
- **Functionality:** This database keeps data on processed proteins as well as the respective embeddings it is engineered to decrease the complexity of information retrieval and ease the process of accessing relevant information promptly during user interactions.

4.1.7 Interaction Flow

- **Process:**
 1. User Query: The user inputs a query through the UI.
 2. API Call: The front end sends the query to the backend via an API call.
 3. Data Retrieval: The information is looked up in the database and embeddings are created where needed by the server in response to queries.
 4. Response Generation: The backend develops a response using the obtained data and embeddings.

```

# manage.py
#!/usr/bin/env python
import os
import sys

def main():
    os.environ.setdefault('DJANGO_SETTINGS_MODULE', 'Daniel.settings')
    try:
        from django.core.management import execute_from_command_line
    except ImportError as exc:
        raise ImportError(
            "Couldn't import Django. Are you sure it's installed and "
            "available on your PYTHONPATH environment variable? Did you "
            "forget to activate a virtual environment?"
        ) from exc
    execute_from_command_line(sys.argv)

if __name__ == '__main__':
    main()

```

Figure 4.1: Manage.py

5. User Response: Then, the reply is transmitted back to the front-end side so that it can be shown on the chat box for the user.

4.2 Development Environment

Django was employed in creating the AI chatbot, a high-level Python web framework that is favored for its desire to accelerate development speed and retain a simple design. The initial set up of the project entailed creating a Django project, setting up prerequisite configurations, as well as establishing applications that would reinforce chatbot functionalities. In this arrangement, Django's manage.py file (Figure 4.1) was keystone because it oversaw different administrative duties like server running, database migration applications besides project settings configurations; Nonetheless, it is important to note how this script assisted in initializing and up keeping development components.

4.3 Data Handling and Storage

Primary data for Chatbot was sourced from the DisProt database (Figure 4.4, Figure 4.5) which is a rich source of information about Disordered Intrinsically Proteins. The data handling entailed several steps hence ensuring that querying and embedding generation was done efficiently.

Data was initially imported from JSON files into the Django application containing various attributes of proteins like names, sequences, and other relevant metadata. Extensive preprocessing was done for the data in order to enable efficient querying as well as the generation of embeddings. This preprocessing involved cleaning data so that it becomes consistent in terms of format removing inconsistencies; normalizing it so that all data are within acceptable criteria; and splitting large texts into smaller parts which makes them easier for most humans or machines to process.

After preprocessing, the data is retained in a PostgreSQL data bank via Django models, wherein each protein entry is kept with its features as well as an embedded vector. The

embedding vector helps present the protein's information in a manner that can be used by machine learning methods. Thus, data could quickly be fetched when users requested it as a result of this manner of storage.

4.4 Embedding Generation

Creating embeddings was crucial for the chatbot's functionality, enabling it to accurately understand and process natural language questions. Embeddings were obtained using the OpenAI API that turned text data into high-dimensional vectors representing semantic information of the words. (Figure 4.2)

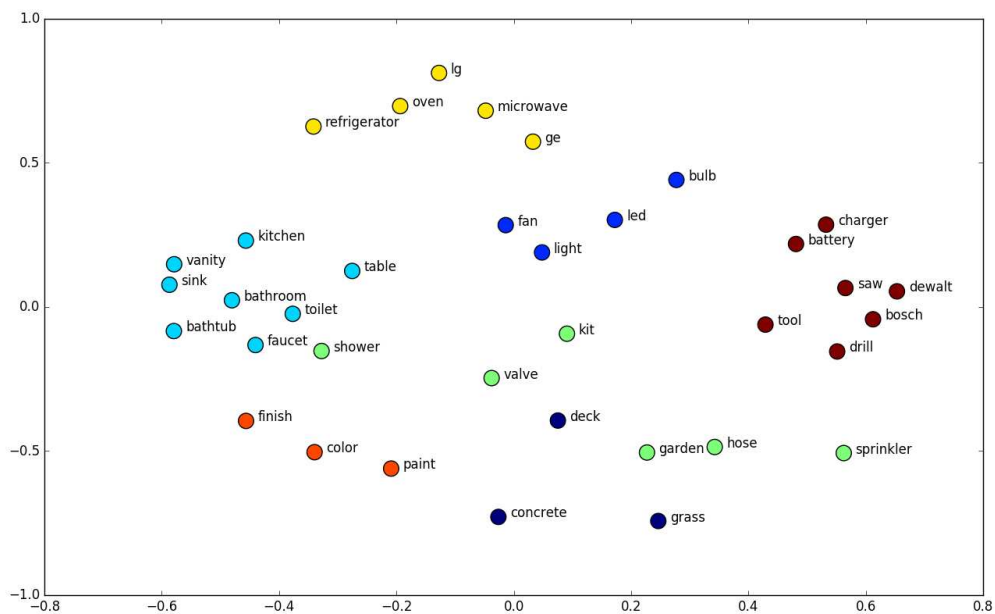


Figure 4.2: Word Embedding - Credit: Neptune Website

To handle large texts, given that the OpenAI API limits the number of tokens per request to 8192, the text was divided into smaller segments using a chunking strategy to avoid exceeding the token limit. This was important for embedding generation as it ensured that the entire text could be processed within the API constraints.

Every chunked text received its own creation of an embedding vector which was then merged to create a single representation. We used the averaging method for integration, which facilitates obtaining a final combination that accurately represents the entire text with all its content reflected in terms of embedding. Thus, enabling the chatbot to efficiently handle large and complex databases while maintaining a higher accuracy rate in its responses.

4.5 API Creation

Several APIs have been created using the Django REST Framework in order to make it easier for people to communicate with chatbots. Through these, creators as well as participants are able to access functionalities from the system programmatically such as creating data, handling inquiries, and fetching data.

Data Creation APIs: Both bulk data imports and single data entries were handled by these APIs. The incoming data was processed, embeddings were generated, and the processed data was stored in the database. The database was thus continuously updated with new and relevant information which allowed the chatbot to ensure that its responses were accurate and up to date.

Query Handling API: This was the central endpoint for our chatbot. Users could send in their questions on individual proteins or ask general ones through it, the natural language processing and machine learning algorithms of this interface could generate relevant replies by processing requests, even if they were formally made as queries. What made this part so important though is that it provides the right answers bearing in mind the user's context whenever they need them.

Data Retrieval API: Additional endpoint in the Data Retrieval API allowed users to download data in CSV format, which could be easily accessed for analysis purposes thus making it suitable even for those scientists or anybody else looking at the proteins other than the chatbot interface.

4.6 Handling Token Limitations and Chunking Data

The OpenAI API has a token limit of 8192 per request, making it difficult to process large texts. This led to splitting the text data into smaller parts. When chunking the data, it was divided into segments that can be easily managed depending on the length of words – so every segment will have fewer than 8192 tokens.

In generating replies, every chunk is treated separately by the chatbot. The chatbot achieves this feat by employing a special function that controls the context and structure of the query, treating each chunk of the document one after the other. Responses from each chunk are combined to make up a complete answer. Through this approach, the chatbot is able to handle even intricate questions, furnishing precise feedback within the token limits.

The chunking strategy, in addition to combining the responses from individual chunks in order to create a unified response, allowed the chatbot to handle large and complex queries efficiently, at the same time as maintaining high accuracy and relevance in its responses.

4.7 User Interface

The main layout was done using HTML, CSS, and JavaScript; it will be easy for everyone to engage in computer conversation since it is made by having user experience best practices in mind. In other words, the user interface was made in such a manner that it is intuitive and easy to use.

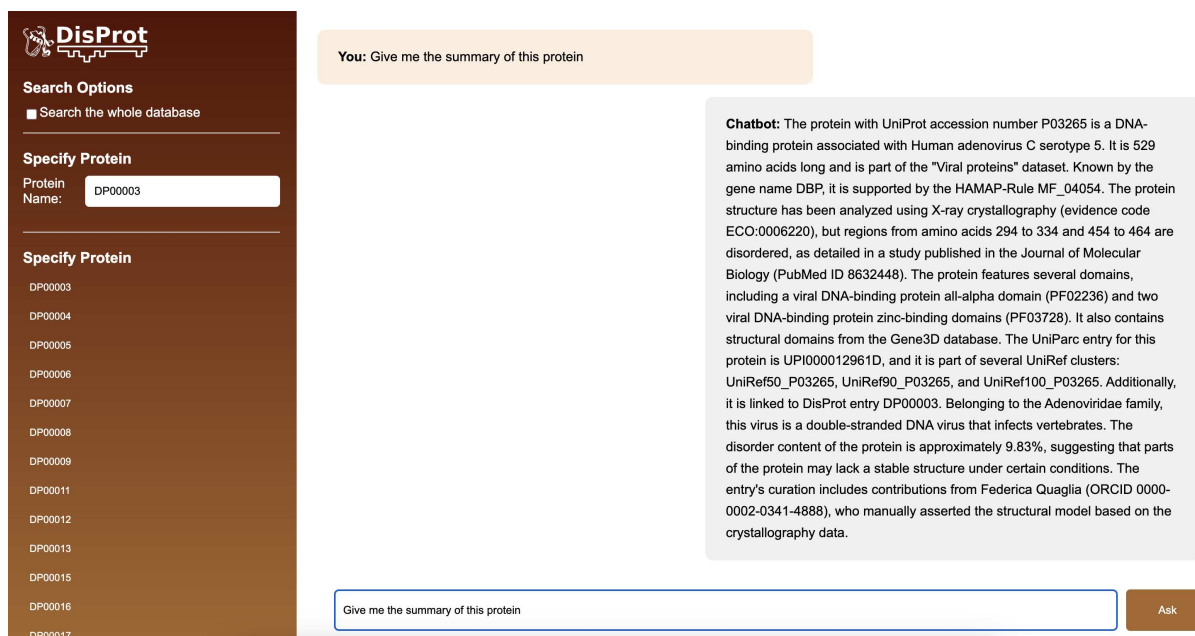


Figure 4.3: Chatbot UI

Layout: In the interface (Figure 4.3), we have a left sidebar which includes different ways to search through the database:

1. Search the Whole Database: Users have the option to look through the complete database.
2. Search by Protein Name: Users have the ability to get answers only for specific proteins just by using the protein's name.
3. Choose from List: If users don't know the protein's name, they can choose the protein from the list.

In the right section, we have an input area where users can send queries to the server:

1. Submit Query: Users enter their query in the input field and submit it.
2. Display Query History: The history of the user's queries is displayed in a chat format after submission.
3. Show Server Response: When the server returns an answer, it will be shown in the chat.

4. **Maintain Interaction History:** The chat keeps a history of all user and chatbot interactions, allowing users to easily review past queries and responses.

This layout ensures that users can easily navigate their query history and receive relevant information promptly.

HTML and CSS: An HTML file structured the layout of the interface. It includes sections for search options as well as protein specification and chat interface. These elements were styled by CSS to create a visually appealing and consistent user experience. Keeping it simple and user-friendly was the cornerstone of its design so that communicating with the chatbot could be done fast/easily by any user who visits the website or app.

JavaScript: JavaScript was used for user interaction. Key functions captured user input, sent queries to the backend API, and displayed responses in the chat interface. The script, which adapted the interface based on user preferences to facilitate narrow or wide searches, was interactive. For instance, users had the option to search the entire database or specify the names of certain proteins. Real-time updates on the user interface like showing search results and chatting box updates with new messages were also taken care of by the JavaScript code.

HTML, CSS, and JavaScript together created a responsive and interactive user interface that was easily usable by anyone inputting queries while chatting with the bot. As queries were typed, instant feedback was provided on the screen, showcasing possible questions, thus attracting users to engage with it. This increased the system's overall functionality by simplifying the necessary processes.

Establishing a solid development environment is part of how the AI chatbot was created, and indeed it performs data management and storage effectively. APIs have been created that enable interaction between different systems or programs and tools like web services work by sending requests over the internet from one application to another. AI chatbot retrieves information using semantic similarity between questions and answers; this operation guarantees some level of relevance whatsoever. Administrative or client relationship management databases were arguably the earliest form of data at all, but today we have them specialized even in the life sciences such as proteomics or genomics.

Figure 4.4: Disprot Website

Figure 4.5: Disprot Website - Protein

Chapter 5

Evaluation and Testing

This chapter explains the testing and evaluation conducted to ensure the chatbot is functional, accurate, and performs excellently. It includes developing a testing methodology, defining performance metrics, conducting tests, and analyzing the results.

5.1 Testing Methodology

The chatbot's performance was measured in realistic situations with the testing procedure focusing on accurate and timely response to user questions. The testing was carried out in three stages i.e., user acceptance testing, integration testing, and unit testing phase.

We conducted unit tests during which individual chatbot components were isolated for assessment, to ascertain that their operations were as intended. These entailed the inspection of the data ingestion and preprocessing modules, embedding generation procedure, and API endpoints. The unit tests were automated and employed Django's incorporated testing framework enabling them to be done consistently over time.

Afterwards, there is an integration test to confirm how the chatbot's parts relate to each other. This phase was about correct data transition within the system covering user requests, processing in the database and then giving responses. Therefore, the integration tests verify that the system works as it should by imitating real user actions.

User acceptance testing involves actual users engaging with the application's functionality by responding to questions on biological topics, which determines its effectiveness. Feedback from this process is crucial for identifying usability errors and establishing better operational standards.

5.2 Performance Metrics

Several performance metrics were defined to quantitatively evaluate the chatbot's performance. These metrics included:

- **Accuracy:** Measured the correctness of the chatbot's responses. Accuracy was evaluated by comparing the chatbot's answers to a set of predefined correct answers

for various queries.

- **Response Time:** We counted how long it took for the chatbot to handle a request and form a reply. This helped make sure that the system was operating effectively and the users were satisfied.
- **Error Rate:** During the testing phases, errors and their frequency and types were tracked. An indication that the system was strong and dependable was the low rate of errors.

5.3 Results and Analysis

The testing process yielded valuable insights into the chatbot’s performance across different metrics.

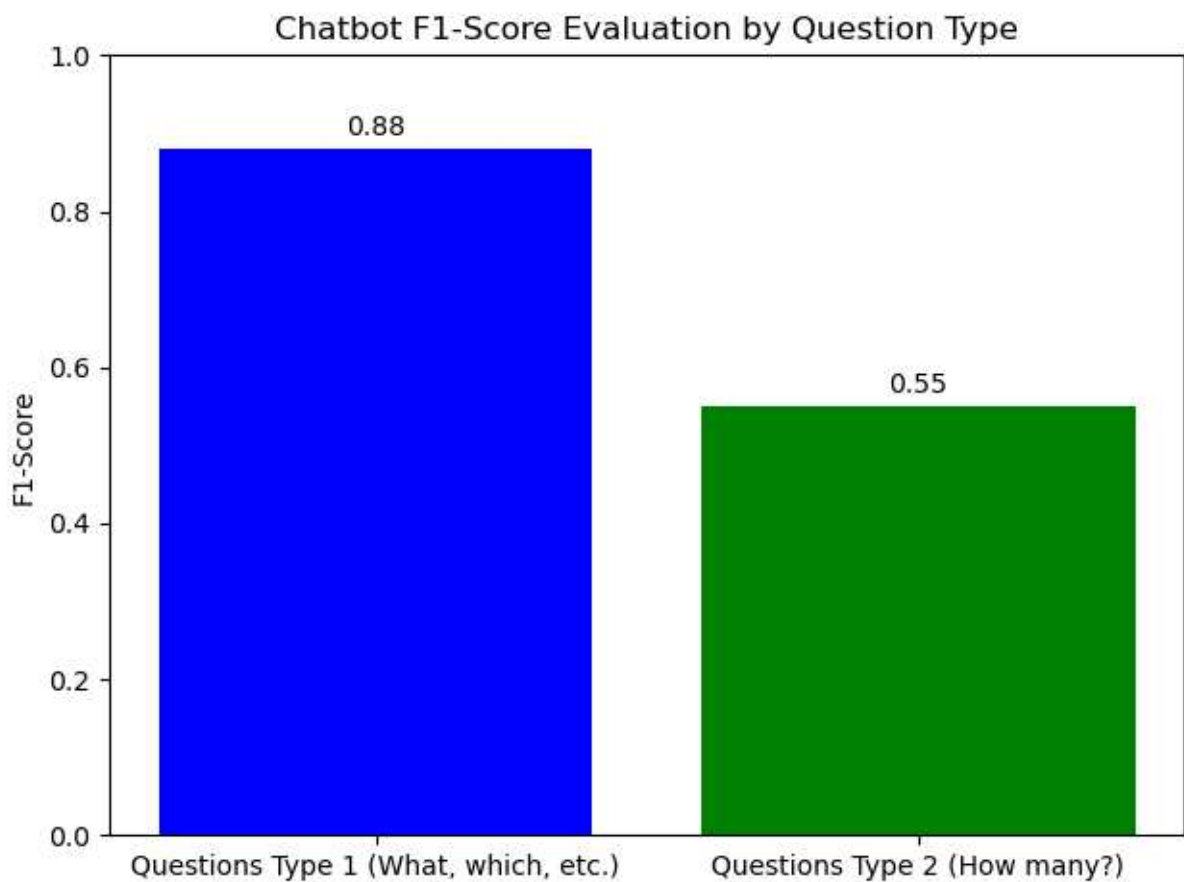


Figure 5.1: F1-Score Plot

Accuracy: A benchmark dataset was created by us containing many queries having the right answers to find out how detailed our chatbot worked. The answers from our chatbot were compared with known right answers from these queries in terms of precision, recall, and F1-score to measure correctness. The chatbot achieved an overall accuracy rate of about 88 percent for questions of type one (which, what, where, etc.) and about 55 percent for questions of type two (how many?), as the results showed, indicating the effectiveness

of the system's embedding generation and natural language processing methods for the type one questions. (Figure 5.1)

Response Time: The chatbot response time for each query and reply logging. One and a half seconds is her typical response time per query in other words she had an average of 6-8 seconds when using *GPT 3.5*, and 12-15 seconds when using *GPT 4o* which was enough for most purposes but complex queries sometimes required more causing delay therefore showing system points which could be optimized.

Error Rate: There were a few errors while testing but most of them were because of edge cases in managing data and some strange ways of asking questions. Most of these problems were solved by having various tests repeatedly hence leading to a strong and dependable system.

5.4 Examples of Chatbot Interactions

Below are some examples of user queries and the corresponding responses generated by the chatbot:

5.4.1 Example 1: Summary of a Protein

Question: Give me the summary of this protein

Chatbot Answer: *"The protein with UniProt accession number P03265 is a DNA-binding protein associated with Human adenovirus C serotype 5. It is 529 amino acids long and is part of the "Viral proteins" dataset. Known by the gene name DBP, it is supported by the HAMAP-Rule MF 04054. The protein structure has been analyzed using X-ray crystallography (evidence code ECO:0006220), but regions from amino acids 294 to 334 and 454 to 464 are disordered, as detailed in a study published in the Journal of Molecular Biology (PubMed ID 8632448).*

The protein features several domains, including a viral DNA-binding protein all-alpha domain (PF02236) and two viral DNA-binding protein zinc-binding domains (PF03728). It also contains structural domains from the Gene3D database. The UniParc entry for this protein is UPI000012961D, and it is part of several UniRef clusters: UniRef50 P03265, UniRef90 P03265, and UniRef100 P03265. Additionally, it is linked to DisProt entry DP00003.

Belonging to the Adenoviridae family, this virus is a double-stranded DNA virus that infects vertebrates. The disorder content of the protein is approximately 9.83 percent, suggesting that parts of the protein may lack a stable structure under certain conditions. The entry's curation includes contributions from Federica Quaglia (ORCID 0000-0002-0341-4888), who manually asserted the structural model based on the crystallography data."

5.4.2 Example 2: Specific Information Request

Question: Hey there, Give the P53 first five evidences?

Chatbot Answer: *"The interaction of p53 with various proteins such as HMGB1, TFIIF, MDM2, CBP, and Tfb1 involves specific regions of p53, leading to structural changes and functional outcomes. Structural studies using techniques like NMR spectroscopy, X-ray crystallography, and isothermal titration calorimetry have provided insights into the binding mechanisms and conformational changes of p53 in complex with its interacting partners. The N-terminal transactivation domain of p53 plays a crucial role in mediating interactions with different proteins, leading to changes in protein structure and function. Post-translational modifications such as acetylation and methylation of p53 residues further modulate its interactions with binding partners and regulate its transcriptional activity. Overall, the combination of structural and biophysical studies has elucidated the molecular mechanisms underlying the interactions of p53 with various proteins and their impact on p53 function."*

5.4.3 Example 3: Disordered Regions in a Protein

Question: Which amino acids are disordered in p53? Just give me the number.

Chatbot Answer: *"Overall, the protein p53 exhibits disorder in multiple regions, including amino acids 1-36, 39-59, 94-312, 367-388, and 370-382. Additionally, specific residues such as 15, 58, 124, 182, 229, 275, and 277 are also disordered. This widespread disorder in p53 may have implications for its function and interactions with other molecules."*

5.4.4 Example 4: Evidence Count for Structural State

Question: How much evidence of structural state is for this protein?

Chatbot Answer: *"Based on the combined responses from the different data chunks provided, there is evidence that the protein alpha synuclein (alpha S) exhibits a disordered structural state in its native form. However, it can undergo conformational transitions to an ordered state, such as an alpha-helical structure, in the presence of certain molecules like SDS micelles or upon interaction with other proteins like tubulin. The evidence for these structural states comes from various experimental techniques, including circular dichroism (CD), nuclear magnetic resonance (NMR) spectroscopy, Fourier transform infrared (FTIR) spectroscopy, and sodium dodecyl sulfate polyacrylamide gel electrophoresis (SDS PAGE). The total number of unique evidences of structural state for alpha synuclein, as derived from the combined responses, is not explicitly provided in the summary, but it is clear that multiple pieces of evidence support the protein's ability to exist in both disordered and ordered states under different conditions."*

5.5 Discussion

The chatbot was able to produce very precise and contextually accurate summaries of protein data. Considering the numerous details contained in a given protein entry, this feature comes in handy. It also handled questions on single proteins very quickly and efficiently

Performance suffered when querying the whole database exhibiting a lack of scalability. Moreover, at 8192 tokens, the current version charges 0.03 dollars per query which means that it is not free for extensive use. Still, there is another limitation with respect to the chunking approach; the accuracy of the system is compromised in cases where the protein exceeds 8192 tokens for questions that require counting (e.g., “how many”).

Although the chatbot has such limitations, it performed well on most query types, which is useful for researchers who need immediate and accurate access to biological data. The goal of future enhancements should be to enhance large-scale query throughput capacity and lower costs related to operations.

Chapter 6

Conclusion

The present section is a summary of the Thesis which includes the advantages of exploiting chatbots enabled by AI in biological databases, limitation identification, and recommendations for unwinding new studies.

6.1 Advantages of AI Chatbots in Bioinformatics

There are several significant advantages when implementing an AI chatbot that accesses biological databases. In addition to facilitating access to complex biological data, even for non-technical users, it also simplified the process of querying such information. The availability of natural language interfaces has made these systems more inclusive and user-friendly.

Besides, the chatbot greatly cuts down on the amount of time and energy that one needs to spend retrieving the needed information. This reduces the time taken to look for relevant information since one can get direct answers within seconds from their queries and this fastens the decision-making process based on data while boosting research productivity more generally. The data utility was improved by making the responses more accurate and grounded in context through the integration of advanced natural language processing and machine learning techniques which allows researchers to derive more meaningful insights. The chatbot's user experience was improved by the use of an intuitive architecture and a user-friendly interface. Consequently, natural language queries made the engagement more interesting and less tedious compared to traditional methods.

6.2 Limitations

Although the project had numerous attendant benefits, it also had various disadvantages. Most queries were handled within acceptable time limits; however, there were some intricate ones that would take longer processing thus necessitating enhancement of mechanisms for retrieving extensive information. The chatbot's accuracy was not entirely reliable. There were times when it did not understand the questions posed by users

especially if they were vague or referred to an extremely narrow subject matter.

Another limitation that influenced directly the chatbot performance was the quality and completeness of the DisProt database. Any inaccuracies in the information undermined how accurate responses were provided by chatbots. To exhaustive or unanticipated queries, the chatbot had limitations because it was limited by available information and typical types of questions that could be expected.

The current implementation has a certain bottleneck in that the expense related to generating embeddings through OpenAI API is 0.03 dollars per 8192 token query which may be high depending on the frequency of usage. Furthermore, it affects responses' correctness to queries that require counting or specific details scattering over various chunks.

6.3 Future Directions of the current chatbot

There are various future directions suggested for developing the capabilities of this chatbot beyond these weaknesses. Among these future directions, one of the main aims is to design a local embedding generator. That means it would be possible to use the system at no cost if it no longer relies on external APIs and minimizes operation expenses by much. The way forward is, therefore, setting up an appropriate framework that includes training models from relevant data sets before attaching them to the current system.

Furthermore, we will attempt to make the system perform better, especially on complicated inquiries and retrieving plenty of information in big data. This can imply modification of data retrieval algorithms and enhancing more efficient data indexing and caching strategies.

Fine-tuning the NLP models for the chatbot continuously will lead to more accurate responses from the bot. In other words, training data sets could have more diverse types of questions and unusual cases listed among them, so that they may become more broad. The chatbot's scope could be expanded by integrating more biological databases so it can better serve researchers. Introducing advanced features like multi-turn conversation management, which enables it to maintain context for several interactions, would make the user experience much better and enable more complex query handling.

Exploring explainable AI techniques can potentially improve the transparency of the chatbot's decision-making process to users. Consequently, users will have a sense of how the chatbot generates answers and trust its reliability.

6.4 Final Thoughts

Designing an AI chatbot to inquire about biological databases has a lot of potential to enhance the availability, speed, and applicability of intricate biological information. However, there are complexities to solve and some issues that need to be addressed, but the benefits of this AI chatbot underscore its possibilities for bioinformatics. Future

developments can further enhance the capabilities and impact of AI chatbots, which helps in making research in biological sciences more efficient and effective. The information from this study will act as a stepping stone for forthcoming innovations and progress in an exciting field that changes rapidly.

References

- [1] Benson DA, Karsch-Mizrachi I, Lipman DJ, Ostell J, Sayers EW. GenBank. *Nucleic Acids Res.* 2009 Jan;37(Database issue).
- [2] Yates A, Akanni W, Amode MR, et al. Ensembl 2020. *Nucleic Acids Res.* 2020 Jan 8;48(D1).
- [3] Kent WJ, Sugnet CW, Furey TS, et al. The human genome browser at UCSC. *Genome Res.* 2002 Jun;12(6):996-1006.
- [4] The UniProt Consortium. UniProt: the universal protein knowledgebase in 2021. *Nucleic Acids Res.* 2021 Jan 8;49(D1).
- [5] Berman HM, Westbrook J, Feng Z, et al. The Protein Data Bank. *Nucleic Acids Res.* 2000 Jan 1;28(1):235-42.
- [6] Mitchell A, Chang HY, Koscielny G, et al. The InterPro protein families database: the classification resource after 15 years. *Nucleic Acids Res.* 2015 Jan;43(Database issue).
- [7] Wishart DS, Feunang YD, Marcu A, et al. HMDB 4.0: the human metabolome database for 2016. *Nucleic Acids Res.* 2016 Jan 4;44(D1).
- [8] Caspi R, Altman T, Dreher K, et al. The MetaCyc pathway database release 21.2. *Nucleic Acids Res.* 2018 Jan 4;46(D1).
- [9] Kanehisa M, Sato Y, Furumichi M, Morishima K, Tanabe M. New approach for understanding genome variations in KEGG. *Nucleic Acids Res.* 2019 Jan 8;47(D1).
- [10] Jassal B, Matthews L, Vedantam S, et al. The reactome pathway knowledgebase. *Nucleic Acids Res.* 2020 Jan 8;48(D1).
- [11] Oughtred R, Stark C, Breitkreutz BJ, et al. The BioGRID interaction database: 2019 update. *Nucleic Acids Res.* 2019 Jan 8;47(D1).

- [12] Szklarczyk D, Gable AL, Lyon D, et al. STRING v11: protein-protein association networks with increased coverage, supporting functional discovery in genome-wide experimental datasets. *Nucleic Acids Res.* 2019 Jan 8;47(D1).
- [13] Manning CD, Raghavan P, Schütze H. *Introduction to information retrieval.* Cambridge: Cambridge University Press; 2008.
- [14] Salton G, Wong A, Yang CS. A vector space model for automatic indexing. *Commun ACM.* 1975 Nov;18(11):613–620.
- [15] Guha, R., McCool, R., Miller, E. (2003). Semantic search. *Proceedings of the 12th International Conference on World Wide Web*, 700-709.
- [16] Wang, Q., Mao, Z., Wang, B., Guo, L. (2017). Knowledge graph embedding: A survey of approaches and applications. *IEEE Transactions on Knowledge and Data Engineering*, 29(12), 2724-2743.
- [17] Jiang, L., Yu, M., Zhou, M., Liu, X., Zhao, T. (2015). Improving semantic relevance for sequence-to-sequence learning of Chinese social media text summarization. *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics*, 1077-1086.
- [18] Lample, G., Ballesteros, M., Subramanian, S., Kawakami, K., Dyer, C. (2016). Neural architectures for named entity recognition. *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 260-270.
- [19] Zeng, D., Liu, K., Lai, S., Zhou, G., Zhao, J. (2014). Relation classification via convolutional deep neural network. *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, 2335-2344.
- [20] See, A., Liu, P. J., Manning, C. D. (2017). Get to the point: Summarization with pointer-generator networks. *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 1073-1083.
- [21] Mitra, B., Diaz, F., Craswell, N. (2017). Learning to match using local and distributed representations of text for web search. *Proceedings of the 26th International Conference on World Wide Web*, 1291-1299.

- [22] Joachims, T. (2002). Optimizing search engines using clickthrough data. Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 133-142.
- [23] Ricci, F., Rokach, L., Shapira, B. (2011). Introduction to recommender systems handbook. In Recommender Systems Handbook (pp. 1-35). Springer.
- [24] Devlin, J., Chang, M. W., Lee, K., Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, 4171-4186.
- [25] LeCun, Y., Bengio, Y., Hinton, G. (2015). Deep learning. Nature, 521(7553), 436-444.
- [26] Gunning, D., Aha, D. W. (2019). DARPA's explainable artificial intelligence (XAI) program. AI Magazine, 40(2), 44-58.
- [27] Ramesh, A., Ravishankaran, S., Joshi, A., Chandrasekaran, K. (2017). A survey of design techniques for conversational agents. Proceedings of the 4th International Conference on Information Systems and Management Science, 51-55.
- [28] Radziwill, N. M., Benton, M. C. (2017). Evaluating the quality of chatbots and intelligent conversational agents. Software Quality Professional, 19(3), 25-36.
- [29] Litman, D., Silliman, S. (2004). ITSPOKE: An intelligent tutoring spoken dialogue system. Demonstration papers at HLT-NAACL 2004, 5-8.