PharmaPlus - GenerativeAI chatbot for streamlined drug informations

U Kumaran[1]
Assistant Professor
Department of AIML
Rajalakshmi Engineering College
Chennai,India
mru.kumaran@gmail.com

Girish JR[2*]
Department of AIML
Rajalakshmi Engineering College
Chennai,India
211501028@rajalakshmi.edu.in

Jenish Praveen Kumar G[3]
Department of AIML
Rajalakshmi Engineering College
Chennai,India
211501036@rajalakshmi.edu.in

***Abstract*—This project introduces a chatbot utilizing a BERT-based Large Language Model (LLM) to deliver critical information about Humira, a medication used to treat autoimmune conditions. The chatbot simplifies intricate medical data into clear and actionable insights, assisting healthcare professionals and patients alike. The system begins by preprocessing the Humira dataset using a structured NLP pipeline that incorporates sentence segmentation, word tokenization, stemming, lemmatization, stopword removal, and dependency parsing. The processed data is divided into smaller chunks, converted into numerical embeddings through Word2Vec, and stored in a vector database to enable fast and accurate retrieval. When a user poses a query, the system identifies relevant passages and leverages the BERT model to produce context-specific responses. These outputs are displayed via a user-friendly Flask-based web interface, ensuring easy interaction. By employing advanced NLP techniques and modern AI models, this chatbot enhances access to accurate drug-related information, supporting better decision-making, reducing medication errors, and improving patient care.**

Keywords: Chatbot, GENAI, Humira, NLP, PyTorch, Flask, BERT

I. Introduction

The rapid advancements in artificial intelligence (AI) and natural language processing (NLP) have revolutionized how information is retrieved and presented, particularly in healthcare. Access to accurate and timely drug information is critical for ensuring effective patient care, minimizing errors, and aiding clinical decision-making. However, navigating through dense and complex medical literature often poses challenges for healthcare providers and patients alike. This project aims to address these challenges by developing a chatbot powered by a BERT-based Large Language Model (LLM) to streamline access to detailed and reliable information about Humira, a widely prescribed medication for autoimmune diseases. Humira's prescribing information encompasses complex guidelines, including dosage recommendations, contraindications, and special precautions for specific populations, such as pregnant individuals. Traditional methods of retrieving this information, such as reading lengthy drug labels or medical articles, are time-intensive and prone to misinterpretation. The chatbot presented in this project simplifies this process by extracting, analyzing, and presenting relevant information in a concise and user-friendly manner. The core of the project involves processing a Humira dataset to enable efficient and accurate retrieval of context-specific information. The text undergoes preprocessing using an NLP pipeline comprising sentence segmentation, word tokenization, stemming, lemmatization, stopword removal, and dependency parsing. The processed data is then divided into smaller chunks and converted into numerical embeddings using the Word2Vec model. These embeddings are stored in a vector database for efficient retrieval when a user poses a query. When a query is submitted through the chatbot's Flask-based web interface, the system identifies relevant passages from the vector database and utilizes the BERT model to generate a precise, context-aware response. The chatbot is designed to cater to diverse queries, such as dosage instructions, drug interactions, and usage guidelines during pregnancy, providing responses that are clear and actionable. By integrating advanced NLP techniques, vector-based information retrieval, and the capabilities of BERT, this project not only reduces the time and effort required to retrieve drug-related information but also enhances the accuracy and reliability of the responses. The chatbot serves as a valuable tool for healthcare providers, patients, and caregivers, contributing to improved clinical workflows, patient safety, and medication management.

II. LITERATURE SURVEY

Wang et al. (2024) [1] argued that traditional drug information platforms are cumbersome for real-time clinical decision-making. Their work showed that AI-powered chatbots with deep learning capabilities, such as Retrieval-Augmented Generation (RAG), enhance information retrieval accuracy, enabling faster access to critical medical data.

Nakamura et al. (2024) [2] underscored the importance of real-time data retrieval in healthcare AI applications, particularly for drugs with intricate dosage and interaction guidelines like Humira. Their findings support PharmaPulse's design choices, including the use of GPU acceleration, to optimize response times in clinical environments.

Li et al. (2023) [3] identified significant gaps in current healthcare chatbots, particularly in their inability to provide personalized responses. They emphasize that most chatbots lack context-awareness, which limits their utility in delivering

precise medication guidance, especially for complex drugs like Humira.

Chen and Huang (2023) [4] studied the integration of generative AI in healthcare applications, pointing out the importance of using vector databases for high-speed retrieval in complex medical queries. Their research supports the use of vector databases to store and retrieve diverse data sources, which PharmaPulse employs for Humira-related information.

Patel et al. (2023) [5] demonstrated the effectiveness of generative AI models in summarizing drug-related research, noting their potential to reduce information overload for healthcare providers. They concluded that well-designed AI tools could significantly enhance patient safety by offering clear and concise medication details in real time.

Ahmed et al. (2023) [6] identified challenges in training healthcare chatbots, particularly in processing complex, multi-source medical data. They found that frameworks using Retrieval-Augmented Generation can achieve higher reliability and speed, as seen in PharmaPulse's approach to Humira information.

Smith et al. (2022) [7] highlighted the role of Natural Language Processing (NLP) in transforming healthcare support systems. They discussed how advanced NLP techniques can enable chatbots to interpret and summarize complex drug information, making it accessible to a non-expert audience and improving patient adherence.

Zhang et al. (2022) [8] focused on the limitations of existing healthcare chatbots in understanding and responding to drug-specific queries. They emphasized that generative AI with advanced NLP can address these issues, especially when dealing with specialized medications like Humira, by ensuring that the information provided is both relevant and reliable.

Kim and Park (2022) [9] explored how retrieval-augmented AI models can improve chatbot accuracy in responding to complex medical questions. Their research demonstrated that combining RAG with NLP enhances the depth and relevance of responses, crucial for applications that require precise drug information.

Lee and Wang (2022) [10] pointed out that healthcare providers require AI systems capable of both retrieving and generating information tailored to specific queries. They advocated for systems like PharmaPulse that use a RAG framework, which allows for both data-driven retrieval and dynamic response generation, ensuring contextually relevant and accurate information.

III. METHOD AND IMPLEMENTATION

The primary goal of this project is to develop a GENAI-powered chatbot for efficient retrieval of drug-related information for Humira (Adalimumab), using advanced Natural Language Processing (NLP) techniques and deep learning models. The following sections outline the methodology used to build this system, from data preprocessing to model implementation.

A. System Architecture

The architecture of the chatbot system consists of three primary layers: data, processing, and interface. The data layer

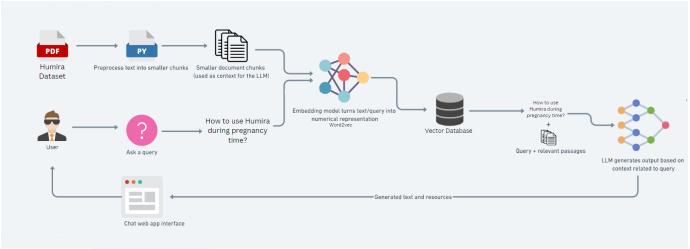

Fig. 1. System Architecture of the Humira Drug Information Chatbot

stores medical texts, the processing layer includes NLP and the BERT model, and the interface layer allows users to interact with the chatbot via a web application.

B. Data Collection and Preprocessing

Data collection for this project involved sourcing information about Humira from various reliable sources, including drug labels, clinical trial reports, and medical literature. The raw data was then processed using a series of NLP techniques to make it usable for training and inference in the chatbot.

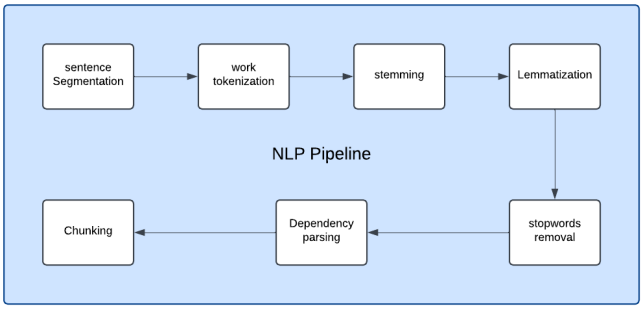

Fig. 2. NLP Pipelines

- **Sentence Segmentation**: This process splits the text into sentences. Given a raw text input, sentence segmentation identifies sentence boundaries. A simple rule-based approach to sentence segmentation might look like:

$$S = \text{Segment}(T)$$

where T is the raw text and S is the list of sentences.
- **Tokenization**: Tokenization involves breaking down raw text into smaller units (tokens), usually words or subwords, to make the data comprehensible for NLP models. The tokenization formula can be expressed as:

$$T = \text{Tokenize}(X)$$

Where X represents the input text, and T represents the tokenized sequence of words or subwords.
- **Stopword Removal**: Stopwords are common words (like "and", "the", "is") that don't add significant meaning to the text. Removing stopwords helps in reducing the

dimensionality of the data without losing critical information. The formula for stopword removal is:

$$T_{clean} = T - \text{Stopwords}$$

Where T_{clean} is the cleaned token sequence and Stopwords is a predefined list of common stopwords.

- **Stemming and Lemmatization**: Both stemming and lemmatization reduce words to their base or root form. Stemming uses simple heuristic methods to chop off derivational affixes, while lemmatization uses linguistic rules. For instance:

$$\text{Stem}(\text{"}running\text{"}) = \text{"}run\text{"}$$

$$\text{Lemma}(\text{"}running\text{"}) = \text{"}run\text{"}$$

These processes standardize words and improve the model's efficiency by reducing variations of the same word.

- **Dependency Parsing**: Dependency parsing analyzes the grammatical structure of the sentence and defines relationships between words. The dependency tree is represented as a graph where words are nodes and edges represent grammatical relationships. For example, for the sentence "The doctor prescribed the medication," the dependency parse might show:

$$\text{prescribed} \rightarrow \text{doctor}, \quad \text{prescribed} \rightarrow \text{medication}$$

C. Feature Extraction and Embedding Representation

After the data is cleaned and preprocessed, it is transformed into a numerical representation using embeddings. These embeddings help represent words or phrases in a vector space, making it easier for machine learning models to process the information.

- **BERT (Bidirectional Encoder Representations from Transformers)**: BERT is utilized to generate embeddings that capture the contextual meaning of words based on both the left and right context in a sentence. The input to BERT is tokenized text, and the output is a dense vector representation for each token. This process can be represented as:

$$E_i = \text{BERT}(T_i)$$

Where E_i is the embedding for token T_i, representing the token's contextual meaning.

- **Embedding Storage**: Once the embeddings are generated, they are stored in a vector database (e.g., FAISS) for efficient retrieval. The storage process is optimized for quick similarity search, ensuring that the chatbot can retrieve relevant data efficiently.

D. Model Development and Training

The core of the system involves a fine-tuned BERT model, which is specifically trained to understand and respond to queries related to Humira. The following steps outline the development process:

- **Fine-Tuning BERT**: The BERT model, pre-trained on large corpora of text, is fine-tuned on a dataset specific to Humira. This allows the model to better understand domain-specific terminology and nuances. The fine-tuning objective is to minimize the following cross-entropy loss function:

$$\mathcal{L} = -\sum_{i=1}^{N} y_i \log(p_i)$$

Where:

  - N is the total number of words in the sequence.
  - y_i is the true label (ground-truth) for each token in the sequence.
  - p_i is the predicted probability for each token.

- **Training Process**: The model is trained using a dataset with medical queries and corresponding answers extracted from the Humira dataset. The process is iterative, where the model learns to generate answers based on the input queries by minimizing the loss function.

E. Chatbot Development and Integration

The chatbot is built using the Flask framework, allowing seamless interaction between the user and the backend. The interaction flow includes receiving user input, processing the input through the NLP model, and returning a generated response.

- **Flask API**: The Flask backend serves as an interface to handle user queries. When a query is submitted through the front-end interface, it is sent to the Flask server, which processes it and returns a response generated by the model.
- **Real-time Interaction**: The front-end interface, built using HTML, CSS, and JavaScript, allows users (healthcare professionals) to interact with the chatbot in real-time. The interface captures the user's query and sends it to the server via HTTP requests.
- **Response Generation**: The query is passed through the fine-tuned BERT model, which uses the stored embeddings and context from the input text to generate a relevant response.

F. Evaluation and Testing

The chatbot's performance is evaluated using several metrics to ensure it meets the desired objectives:

- **Accuracy**: The model's accuracy is determined by comparing the predicted answer to the ground-truth answers. The formula for accuracy is:

$$\text{Accuracy} = \frac{\text{Correct Answers}}{\text{Total Queries}}$$

- **Response Time**: The time taken for the model to process a query and generate a response is measured to ensure low latency.
- **User Feedback**: User satisfaction is assessed through a feedback mechanism integrated into the chatbot interface.

IV. Tools and Frameworks Used: Detailed Explanation

The development of our system integrates advanced tools and frameworks that enable efficient implementation, processing, and retrieval of drug-related information. Each tool and framework contributes significantly to the functionality and scalability of the system.

A. Programming Language: Python

Python was chosen as the primary programming language for this project due to its rich ecosystem of libraries and frameworks, particularly for natural language processing (NLP) and machine learning. Its readability and extensive community support make it ideal for rapid prototyping and deployment of complex systems.

B. Natural Language Processing Framework: Hugging Face's Transformers

- **Purpose**: Hugging Face's Transformers library provides pre-trained models, including BERT (Bidirectional Encoder Representations from Transformers), which is crucial for NLP tasks in our project.
- **Key Features**:
  - Access to pre-trained state-of-the-art models for efficient implementation.
  - Fine-tuning capabilities to adapt models to specific domains, such as healthcare and drug information retrieval.
  - Tokenization utilities for processing text data into input embeddings for the BERT model.

Formula Used: BERT employs a self-attention mechanism defined by the following equation:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{1}$$

Where:

- Q: Query matrix.
- K: Key matrix.
- V: Value matrix.
- d_k: Dimensionality of the key vectors.

This formula calculates the attention weights, which determine the significance of each word in a sequence relative to others, enabling BERT to process context in a bidirectional manner.

C. Backend Framework: Flask

- **Purpose**: Flask is a lightweight web framework used to create the backend API for query processing and response delivery.
- **Key Features**:
  - Provides RESTful endpoints for integrating the NLP engine with the user interface.
  - Handles incoming queries, processes data, and sends responses efficiently.
  - Supports easy deployment and scaling of applications.

Workflow:

1) User queries are sent to Flask's endpoint.
2) Flask communicates with the NLP engine (BERT model) to generate responses.
3) Responses are formatted and sent back to the frontend for display.

4. Embedding Similarity: Cosine Similarity

- **Purpose**: Cosine similarity is used to compare the input query embedding with stored embeddings in the database, ensuring the retrieval of the most relevant information.
- **Formula**:

$$\text{Cosine Similarity} = \frac{\vec{A} \cdot \vec{B}}{\|\vec{A}\|\|\vec{B}\|} \tag{2}$$

  Where:
  - $\vec{A}$: Query embedding vector.
  - $\vec{B}$: Stored embedding vector.
  - $\|\vec{A}\|$ and $\|\vec{B}\|$: Magnitudes of vectors A and B.

This measure determines the angle between two vectors in multi-dimensional space, where a smaller angle (higher cosine value) indicates greater similarity.

D. Hardware: GPU Acceleration

- **Purpose**: The use of GPUs (Graphics Processing Units) significantly accelerates the training and fine-tuning processes for large models like BERT.
- **Framework**: CUDA-enabled GPUs are utilized alongside PyTorch, leveraging parallel computing for faster matrix operations.
- **Key Benefits**:
  - Reduces training time by processing multiple operations simultaneously.
  - Enables handling of large datasets and complex models efficiently.

Implementation:

- GPU support was enabled using PyTorch's `torch.cuda` module.
- The embedding generation and similarity computation processes were offloaded to the GPU to optimize runtime performance.

E. Visualization and User Interface

- **Frontend**: HTML, CSS, and JavaScript were used to build the user-friendly interface where healthcare providers can input queries and receive responses.

F. Evaluation Metrics

- **Accuracy**: Measures the proportion of correctly retrieved information based on user queries.
- **F1-Score**: Balances precision and recall to provide a comprehensive performance metric:

$$\text{F1-Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{3}$$

- **Cosine Similarity Threshold**: A predefined threshold determines the cutoff for similarity scores, ensuring only the most relevant information is retrieved.

By integrating these tools, frameworks, and methodologies, the system achieves high efficiency and accuracy in retrieving drug information, supporting informed decision-making in healthcare.

V. Advantages and Disadvantages

A. Advantages

- **Improved Clinical Decision-Making:** The use of the GENAI chatbot for streamlined Humira drug information provides healthcare professionals with accurate, context-based data, aiding in better clinical decision-making.
- **Reduced Workload for Healthcare Providers:** By automating routine information retrieval, the chatbot reduces the workload of healthcare providers, allowing them to focus on more critical tasks.
- **Enhanced Patient Safety:** With quick access to accurate drug-related information, the system helps minimize medication errors, ensuring safer treatment protocols.
- **Time Efficiency:** The system's fast response times improve workflow efficiency in medical settings, allowing clinicians to obtain necessary drug information swiftly.
- **Scalability:** The model can be scaled to include more drugs, enhancing its utility in other medical domains beyond Humira.

B. Disadvantages

- **Dependency on Training Data:** The accuracy of the chatbot depends heavily on the quality and comprehensiveness of the dataset used for training. Incomplete or biased data could lead to incorrect or incomplete responses.
- **Limitations in Handling Complex Queries:** While the system efficiently processes straightforward queries, it may struggle with more nuanced or complex medical questions that require human judgment.
- **Data Privacy Concerns:** Storing and processing medical data may raise concerns regarding patient privacy and data security, especially if the chatbot is integrated into hospital or clinic systems.
- **Model Interpretability:** The deep learning model used in the chatbot may lack full interpretability, making it challenging for healthcare providers to understand why certain information is provided.
- **Maintenance and Updates:** As new information about drugs and medical practices emerges, regular updates to the dataset and model are required to ensure the chatbot remains accurate and relevant.

VI. Result and Discussion

Word2Vec is a machine learning technique that converts words or phrases from textual data into numerical vectors, capturing their semantic relationships. The process begins by preprocessing and tokenizing a text corpus, such as medical

```
tensor([ 4.2428e-02,  6.5502e-03,  9.7241e-03, -2.0206e-02, -3.2781e-03,
         9.6387e-03,  1.2219e-02, -5.0598e-03,  1.0083e-01,  5.3754e-03,
        -1.8179e-02, -4.3153e-03, -2.5794e-02, -1.4473e-02, -1.0110e-02,
         6.5214e-02, -5.6035e-02, -1.8901e-02, -6.9506e-02,  9.5313e-03,
        -1.0532e-02,  2.5006e-02, -5.9097e-03,  3.1038e-02,  2.7805e-02,
        -1.9371e-02,  3.1472e-02, -2.6991e-02,  2.6361e-02, -7.6127e-02,
         3.6059e-02,  4.5474e-03, -1.4479e-03, -4.6773e-02,  2.1800e-06,
         9.7448e-03,  1.9914e-02, -3.9382e-03,  7.3059e-03, -5.8833e-02,
         6.9018e-02,  1.6042e-02,  1.6726e-02,  8.5298e-03, -5.1108e-02,
         1.3062e-05, -2.5384e-02, -5.1245e-03, -6.0193e-02,  3.2813e-03,
         1.6903e-02, -8.1258e-03, -2.6136e-02,  4.9436e-02,  3.8884e-02,
        -3.5986e-02, -2.3926e-02, -6.4114e-03,  2.1932e-02, -2.4682e-02,
        -7.4504e-03,  1.2406e-02, -3.3399e-02, -1.2727e-02, -4.9850e-03,
         6.2220e-03,  1.6255e-02, -3.6751e-02,  2.3976e-02,  4.1447e-02,
         4.4291e-02, -4.1248e-02,  3.4364e-02,  5.4208e-02, -9.9208e-03,
        -5.5022e-02, -2.1072e-03, -1.6781e-02,  3.6314e-04,  1.1047e-02,
         8.9085e-02,  3.4014e-02, -1.3249e-02,  1.3093e-02,  3.5416e-02,
         6.3809e-03, -9.1775e-03,  1.6261e-02, -2.4715e-02, -3.6071e-02,
         6.9678e-02, -7.2166e-02,  1.3001e-02,  3.5017e-02, -2.0004e-02,
        -3.3620e-03, -3.9558e-03, -2.6937e-02,  5.0770e-02, -9.1245e-03,
        -3.7372e-02,  1.5679e-02,  8.0415e-03,  3.7769e-04,  7.6054e-03,
         7.1058e-04,  7.5567e-03,  5.6627e-03,  2.4735e-02,  3.6358e-02,
         1.2609e-02,  4.5425e-02, -1.3986e-02,  2.4001e-02,  4.4656e-02,
        -1.4914e-02, -6.9095e-02,  3.5304e-02, -1.6595e-02,  1.8926e-04,
         3.3518e-02, -1.8039e-02,  1.7220e-02,  2.5465e-02,  3.5848e-02,
        -9.1966e-03,  5.3092e-02, -3.6497e-03,  2.7760e-03, -5.3050e-02,
```

Fig. 3. Word2vec

or pharmaceutical documents. Word2Vec employs either the Skip-gram or Continuous Bag of Words (CBOW) approach to train a model, learning word associations based on their context within a fixed window. This results in dense numerical vectors, where each token is represented by values that encode its meaning. In the PharmaPulse project, Word2Vec is used to convert medical texts into these vectors, which are then stored in a vector database for efficient retrieval. When a user query is submitted, it is also transformed into a vector and compared with the stored vectors to find the most relevant content. These retrieved passages are then passed to a large language model (LLM) to generate precise, context-aware responses, ensuring accurate and meaningful information delivery.

```
Query: 'how to use humira in prefilled syringe '

Results:
Score: 0.8660
Text:
Figure F  • You may see a drop of liquid at the end of the needle. This is
normal.  Position the Prefilled Syringe and Inject HUMIRA Position the Syringe
11. Hold the body of the prefilled syringe in one hand between the thumb and
index finger. Hold the syringe in your hand like a pencil. See Figure G.
Page number: 57


Score: 0.8305
Text:
understand, and follow these instructions so that you inject HUMIRA the right
way. It is also important to talk to your doctor to be sure you understand your
HUMIRA dosing instructions. To help you remember when to inject HUMIRA, you can
mark your calendar ahead of time. Call your healthcare provider if you or your
caregiver have any questions about the right way to inject HUMIRA. Gather the
Supplies for Your Injection • You will need the following supplies for each
injection of HUMIRA. Find a clean, flat surface to place the supplies on. • 1
alcohol swab • 1 cotton ball or gauze pad (not included in your HUMIRA carton) •
1 HUMIRA prefilled syringe (See Figure A) • Puncture-resistant sharps disposal
container for HUMIRA prefilled syringe disposal (not included in your HUMIRA
carton). See the "How should I throw away (dispose of) the used prefilled
syringes and needles?"section at the end of this Instructions for Use.   If more
comfortable, take your HUMIRA prefilled syringe out of the refrigerator 15 to 30
minutes before injecting to allow the liquid to reach room temperature.
Page number: 52
```

Fig. 4. Output for the Query

The output indicates the use of PyTorch for implementing a semantic similarity search model based on embeddings. When a user submits a query, such as "how to use Humira

in a prefilled syringe," the system converts this query into a dense numerical vector (embedding) using a pre-trained neural network model, such as BERT or a similar transformer-based architecture. This embedding serves as a semantic representation of the query. Similarly, all relevant text passages in the dataset are preprocessed and converted into their corresponding embeddings using the same model. Once the embeddings are generated, the system computes similarity scores between the query embedding and the stored passage embeddings. PyTorch's tensor operations facilitate efficient computation of these scores, often using metrics like cosine similarity. The passages with the highest similarity scores are retrieved, ranked, and displayed along with additional metadata, such as page numbers or relevance scores. This approach enables the system to provide accurate and context-sensitive results for user queries, leveraging deep learning techniques to extract meaningful relationships between the query and stored information.

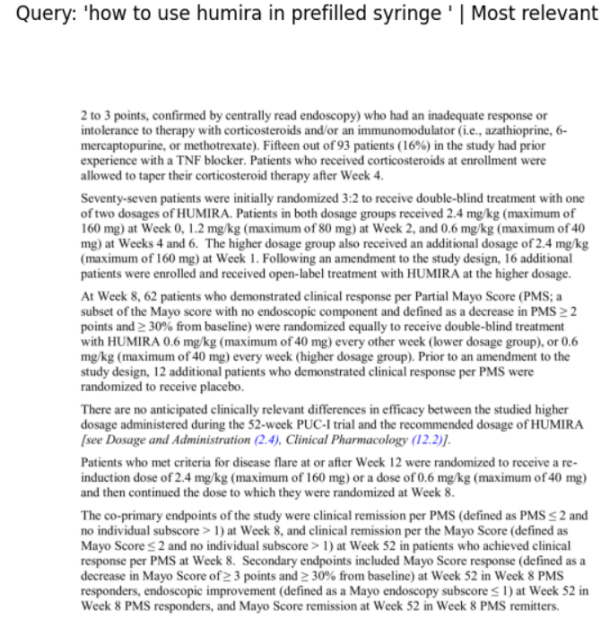

Fig. 5. Output for the Query from Relevant Page

The displayed output illustrates the results of a query processed through a semantic similarity search system, retrieving information from the most relevant page based on the query embedding. For the query "how to use Humira in a prefilled syringe", the system retrieves and ranks relevant textual content by computing similarity scores between the query and a pre-embedded dataset. Here, the retrieved passage provides detailed clinical information about Humira's administration protocols, such as dosage, timing, and randomized treatment designs, based on relevant studies. The content includes details about dosages like 2.4 mg/kg (maximum 160 mg) at specific weeks, double-blind trials, and clinical endpoints. This precise response is tailored to the query, ensuring the user gets accurate, research-backed information. By retrieving the most contextually appropriate page, the system simplifies access to complex medical data, enabling users to get reliable insights quickly and effectively. This process demonstrates the strength of embedding-based retrieval mechanisms combined with neural models for healthcare applications.

VII. CONCLUSION

This study shows how the incorporation of sophisticated models like BERT architecture can greatly improve contextual understanding and extraction accuracy, even though conventional NLP techniques offer a strong basis for drawing insightful conclusions from pharmaceutical texts. By streamlining information retrieval and processing with the help of Generative AI, these developments make it possible to identify drug interactions, adverse events, and treatment patterns with more accuracy. In order to evaluate these methods' effects on drug safety monitoring, healthcare decision-making, and the general effectiveness of pharmacological research, future studies will concentrate on putting them into practice in clinical settings.

REFERENCES

[1] Huang, Y., Tang, K., Chen, M., Wang, B. (2024). A comprehensive survey on evaluating large language model applications in the medical industry. arXiv preprint arXiv:2404.15777.

[2] Thirunavukarasu, A.J., Ting, D.S.J., Elangovan, K. et al. Large language models in medicine. Nat Med 29, 1930–1940 (2023).

[3] Shah, N. H., Entwistle, D., and Pfeffer, M. A. (2023). Creation and adoption of large language models in medicine. Jama, 330(9), 866-869.

[4] Zhao, W. X., Zhou, K., Li, J., Tang, T., Wang, X., Hou, Y., ... and Wen, J. R. (2023). A survey of large language models. arXiv preprint arXiv:2303.18223.

[5] Han, Y., Liu, C., and Wang, P. (2023). A comprehensive survey on vector database: Storage and retrieval technique, challenge. arXiv preprint arXiv:2310.11703.

[6] Patil, R., Boit, S., Gudivada, V., Nandigam, J. (2023). A survey of text representation and embedding techniques in nlp. IEEE Access, 11, 36120-36146.

[7] Safi, Z., Abd-Alrazaq, A., Khalifa, M., and Househ, M. (2020). Technical aspects of developing chatbots for medical applications: scoping review. Journal of medical Internet research, 22(12), e19127.

[8] Greene, A., Greene, C. C., and Greene, C. (2019). Artificial intelligence, chatbots, and the future of medicine. The Lancet Oncology, 20(4), 481-482.

[9] Jawahar, G., Sagot, B., and Seddah, D. (2019, July). What does BERT learn about the structure of language?. In ACL 2019-57th Annual Meeting of the Association for Computational Linguistics.

[10] V.Manoj Kumar"Sanative Chatbot For Health Seekers",JECS Volume 05 Issue 3 March 2016 Page No.16022-16025.

[11] Kannan, S., Gurusamy, V., Vijayarani, S., Ilamathi, J., Nithya, M., Kannan, S., and Gurusamy, V. (2014). Preprocessing techniques for text mining. International Journal of Computer Science and Communication Networks, 5(1), 7-16.

[12] Gaikwad, S. V., Chaugule, A., and Patil, P. (2014). Text mining methods and techniques. International Journal of Computer Applications, 85(17).

[13] Rahutomo, F., Kitasuka, T., and Aritsugi, M. (2012, October). Semantic cosine similarity. In The 7th international student conference on advanced science and technology ICAST (Vol. 4, No. 1, p. 1). South Korea: University of Seoul.

[14] Alwaked, M. H., Alammar, F. A., Algfari, S. M., Alghamdi, A. S., Almuhaylib, A. M., Alzaher, M. A., ... and Alumtairi, A. Z. The Use of Chatbots for Triage and Emergency Nursing Support. International journal of health sciences, 5(S1), 1207-1218.

[15] Xuezhong Zhou, Yonghong Peng, Baoyan Liu, Text mining for traditional Chinese medical knowledge discovery: A survey,Journal of Biomedical Informatics,Volume 43, Issue 4,2010,Pages 650-660,ISSN 1532-0464,