**Scientific Report: Design and Implementation of a Context-Aware Medical Chatbot for Disease Symptom Extraction Using LLMs and Semantic Search**

**Abstract**

This project presents the development of a context-aware medical chatbot designed to extract and summarize disease symptoms from a structured medical dataset. Leveraging a combination of semantic search, sentence embeddings, and a large language model (LLM), the system performs Retrieval-Augmented Generation (RAG) to provide factually grounded responses. Additionally, the pipeline includes automated regex-based symptom extraction and structured data storage. The chatbot demonstrates efficient and scalable processing of medical texts, offering potential applications in clinical decision support systems (CDSS) and healthcare knowledge bases.

**1. Introduction**

The use of AI-driven chatbots in healthcare is increasingly valuable for assisting both patients and medical professionals. However, LLMs often suffer from hallucination, raising reliability and safety concerns in sensitive domains like medicine. This project aims to mitigate these risks by combining dense vector retrieval (semantic search) with controlled prompting to generate reliable, context-aware responses about disease symptoms and remedies.

**2. Methodology**

**2.1 Data Preparation**

Medical textual data for multiple diseases was provided in a zipped format (data.zip). The pipeline:

* Extracts and organizes data by disease.
* Reads content.txt files containing structured information about symptoms, remedies, and disease descriptions.

**2.2 Text Splitting and Embedding**

* Each disease's content is tokenized and chunked using **LangChain's RecursiveCharacterTextSplitter**.
* Sentence embeddings are generated using the **SentenceTransformer (distiluse-base-multilingual-cased-v2)** model.
* Chunks are stored in **ChromaDB** for efficient similarity search.

**2.3 Semantic Search and Retrieval**

* For each disease, a query is encoded into embedding space.
* The **top 3 semantically relevant chunks** are retrieved from ChromaDB to provide context for the LLM.

**2.4 Prompt Engineering and Controlled Generation**

* A structured prompt template instructs the LLM to:
  + Answer truthfully based on context.
  + Return "No Match" if insufficient information is present.
* The quantized **Mistral-7B-Instruct-v0.3** model is used for generation, improving memory efficiency.

**Example Query:**

sql

CopyEdit

What are the symptoms and remedies of the disease acute\_urticaria?

**2.5 Regex-based Symptom Extraction**

* The model’s output is post-processed using regex to extract the **"Symptoms:"** section:

regex

CopyEdit

Symptoms:\s\*(.\*?)\s\*(?=Remedies:|$)

* This ensures that only medically relevant data is saved.

**2.6 Automated Batch Processing and Storage**

* A loop iterates through all diseases, processes queries, and stores extracted symptoms in a structured folder hierarchy:

swift

CopyEdit

/content/main\_data/{disease\_name}/{disease\_name}\_symptoms.txt

**3. Results**

* The system successfully processed **all diseases** in the dataset.
* Extracted symptoms were stored in structured text files, ready for downstream use (dashboards, knowledge bases, etc.).
* Regex post-processing effectively filtered out unrelated model output, ensuring medical relevance.

**Sample Extracted Output:**

diff

CopyEdit

Disease: Acute Urticaria

Symptoms:

- Sudden onset of raised, red welts (hives)

- Itching, burning, or stinging sensations

- Swelling (angioedema) in some cases

**4. Discussion**

**4.1 Strengths**

✅ **Context-Aware Generation:** Answers were strictly based on retrieved verified content.  
✅ **Safety Mechanism:** The "No Match" fallback ensured hallucination prevention.  
✅ **Scalability:** Automated processing supports large datasets efficiently.  
✅ **Model Efficiency:** 4-bit quantization allowed efficient LLM use on limited hardware.

**4.2 Limitations**

* Regex dependency might miss symptoms if content is poorly formatted.
* The LLM's reliance on prompt context limits its ability to answer general medical queries outside the dataset.
* Further validation with medical experts is required for clinical deployment.

**5. Future Work**

* **Knowledge Graph Construction:** Extend symptom extraction to build disease-symptom graphs.
* **EMR Integration:** Connect the chatbot with Electronic Medical Records for real-time decision support.
* **Explainability Layer:** Include model reasoning trace for medical validation.

**6. Conclusion**

This project demonstrates a robust pipeline combining semantic search, LLMs, and regex-based extraction for creating a medical chatbot capable of extracting disease symptoms accurately. The methodology balances performance and safety, setting the foundation for clinical AI tools focused on trustworthy and explainable outputs.

**7. References**

1. Reimers, N., & Gurevych, I. (2019). Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks.
2. HuggingFace Transformers and Sentence-Transformers Libraries
3. LangChain Documentation
4. ChromaDB Documentation
5. Mistral-7B-Instruct-v0.3 Model Card - Hugging Face