Mediknow - A Malayalam Cancer Question Answering System

Anjali Rajendran
Department of Computer Science and Engineering
Sree Buddha College of Engineering, Pattoor
Alappuzha, Kerala, India
anjalijrajendran@gmail.com

Vijay Biju
Department of Computer Science and Engineering
Sree Buddha College of Engineering, Pattoor
Alappuzha, Kerala, India
vijaybiju234@gmail.com

Alex G Daniel
Department of Computer Science and Engineering
Sree Buddha College of Engineering, Pattoor
Alappuzha, Kerala, India
alexgdaniel201@gmail.com

Sabari Krishna R
Department of Computer Science and Engineering
Sree Buddha College of Engineering, Pattoor
Alappuzha, Kerala, India
rsabarisai2000@gmail.com

Dhanunath R, Assistant Professor
Department of Computer Science and Engineering
Sree Buddha College of Engineering, Pattoor
Alappuzha, Kerala, India
dhanunath.nath@gmail.com

***Abstract* - This paper introduces "MediKnow," a pioneering Malayalam Question Answering System designed to address the scarcity of generative answer works in the realm of healthcare information accessibility, specifically tailored for cancer-related queries. The dearth of such systems in Dravidian languages, particularly Malayalam, has motivated the development of a robust solution. Leveraging advanced Natural Language Processing (NLP) techniques, including OpenAI models and FAISS for efficient vector storage, MediKnow employs a specialized Malayalam language model to navigate the intricacies of the Dravidian linguistic context. The processing pipeline encompasses document loading, text splitting, and embeddings, enhancing the system's capacity to comprehend and accurately respond to a diverse range of cancer-related questions. This work underscores the critical need for bridging the gap in generative answer works for Dravidian languages, highlighting the specific challenges posed by the Malayalam language due to its complexity. Beyond providing accessible information, MediKnow exemplifies the efficacy of employing state-of-the-art NLP technologies to address linguistic nuances. The paper evaluates the system's performance on a dataset of cancer-related questions, demonstrating its ability to deliver accurate and informative answers. The innovative approach presented herein contributes to the advancement of NLP capabilities in non-English languages, particularly focusing on healthcare-related information retrieval. The development and deployment of "MediKnow" signify a significant stride in tackling linguistic and domain-specific challenges in cancer-related question answering, ultimately making critical healthcare information more accessible to Malayalam speakers.**

Keywords: ***Natural Language Processing, Question Answering System, Dravidian Languages, Cancer Information, OpenAI, Faiss.***

I. INTRODUCTION

The accessibility of information plays a crucial role in healthcare, particularly for sensitive topics like cancer. However, existing resources for Malayalam speakers seeking cancer-related information are often limited or inaccessible. This paper introduces MediKnow, a Malayalam question-answering system designed to address this gap. MediKnow leverages state-of-the-art generative language models to provide comprehensive answers to user queries in their native language. This significantly improves accessibility for Malayalam speakers, empowering them with accurate and essential information about cancer. The field of medical information retrieval often focuses on English and other widely spoken languages. This creates a language barrier for individuals seeking information in their native tongue, particularly in Dravidian languages like Malayalam. Accessing reliable and accurate health information becomes a greater challenge, leading to potential misinformation and delayed access to critical knowledge regarding their condition. MediKnow fills this void by offering a user-

friendly interface for Malayalam speakers to ask questions about cancer and receive informative responses. The system utilizes a multi-step approach combining advanced language models and search techniques to retrieve relevant information from trusted sources. This process ensures the accuracy and trustworthiness of the answers provided, empowering users with reliable information for informed decision-making.

A. Problem Statement

In the context of the Malayalam-speaking population, there is a noticeable gap in the availability of comprehensive cancer-related information in their native language. This linguistic barrier restricts individuals from obtaining timely and reliable answers to their cancer-related queries, thereby impeding informed decision-making, early detection, and prevention efforts.

B. Research Objectives

The system aims to achieve this through three key objectives:

- **RO1.** By offering a platform in Malayalam, it eliminates the language barrier and expands access to essential knowledge.
- **RO2.** MediKnow utilizes advanced techniques and trusted sources to create response text that is factually accurate and aligns with current medical knowledge. This ensures users receive reliable and trustworthy information.
- **RO3.** The system prioritizes user-friendliness and intuitiveness. It allows users to interact in their native language, providing a comfortable and familiar experience.

C. Contributions

The contributions of this paper to the field of Malayalam question answering systems and healthcare information access:

- MediKnow specifically caters to the Malayalam-speaking population, bridging the language barrier and providing access to cancer-related information in their native language.
- MediKnow empowers Malayalam-speaking individuals to seek and receive accurate and up-to-date information about cancer, facilitating informed decision-making and promoting better health outcomes.
- MediKnow demonstrates the potential of generative models and natural language processing techniques in developing effective question answering systems for specific domains, such as healthcare.

II. RELATED WORKS

In the domain of natural language processing (NLP) and question answering systems, several notable works have been undertaken, addressing specific linguistic and application challenges.

Antony et al. (June 2019) presented a significant contribution by developing a Support Vector Machine (SVM) based Part of Speech (POS) tagger tailored for Malayalam, a Dravidian language [6]. The authors meticulously addressed the complexities of Malayalam grammar, introducing a specific tagset and demonstrating the effectiveness of SVM in POS tagging through systematic evaluations.

Renjit and Idicula (May 2021) delved into Natural Language Inference (NLI) for Malayalam, exploring various embedding approaches such as Doc2Vec, fastText, BERT, and LASER [7]. Their thorough experimental evaluation, error analysis, and comparison with baseline methods, particularly emphasizing the language-agnostic nature of LASER embeddings, contribute significantly to the literature on NLI, especially for low-resource languages.

Faria et al. (August 2023) explored the application of OpenAI's GPT-3 for Question Answering (QA) over Linked Data (LD), focusing on SPARQL query generation [8]. The study provides insights into the strengths and limitations of using GPT-3 for generating SPARQL queries, contributing to the broader field of QA over Knowledge Graphs.

In another study, Bibin and Anto (Jan 2019) addressed the challenges of building a Question Answering System (QAS) for Malayalam, specifically focusing on question classification using SVM [9]. Their work demonstrated the effectiveness of SVM in classifying Malayalam questions, emphasizing its critical role in developing accurate QAS.

These related works collectively contribute to advancing NLP capabilities in Dravidian languages, particularly Malayalam, addressing challenges in POS tagging, NLI, QA over Linked Data, and question classification. Each study provides valuable insights and methodologies that can inform and complement the development of advanced language models and question answering systems in non-English languages, contributing to the broader goal of making critical information more accessible in diverse linguistic contexts.

III. METHODOLOGY

A. Dataset Design

The dataset designed for the "MediKnow" is a crucial component, containing information about various types of cancer in the Malayalam language. The structure is organized to ensure comprehensive coverage of cancer-related topics, allowing the system to deliver accurate and contextually relevant answers.

a) Content: Each document in focuses on a specific type of cancer, ensuring a granular understanding of individual diseases. Information includes symptoms, causes, severity, treatments, and other relevant details specific to each cancer

type. Covers diverse aspects of cancer, addressing the user's potential queries comprehensively. Encompasses both general information and specific details related to the impact of different cancers.

b) Language Considerations: Accounts for linguistic nuances in Malayalam script and vocabulary. Specialized preprocessing addresses the challenges posed by the unique linguistic characteristics of Malayalam. Accommodates variations in Malayalam language usage across regions and communities. Enhances the system's adaptability to a diverse user base.

c) Training Set: Used to train language models and embeddings. Ensures that the system learns from a diverse range of cancer-related information in Malayalam.

d) Testing Set: Employed to evaluate the system's performance. Assesses how well the system can generate accurate and contextually relevant answers for unseen data.

B. Document Splitting

Document splitting, often referred to as chunking, is a preprocessing step in natural language processing (NLP) that involves breaking down lengthy documents into smaller, more manageable chunks. This is particularly useful when dealing with large texts or documents, ensuring that the subsequent analysis is efficient and scalable.

RecursiveCharacterTextSplitter to split lengthy documents into smaller, manageable chunks. The RecursiveCharacterTextSplitter operates by recursively dividing a document into chunks based on specified parameters. It intelligently identifies natural breaks in the text to create meaningful chunks. Optimization of Parameters: Parameters such as chunk size and overlap are crucial for optimizing the document splitting process. Determines the maximum length of each chunk. Balancing between too small (losing context) and too large (inefficient analysis) is essential. Specifies the number of characters shared between adjacent chunks. Overlapping ensures that important contextual information is not missed. The optimal chunking parameters depend on the characteristics of the documents in the dataset. Factors such as the average length of paragraphs, presence of headers, or common document structures influence the choice of parameters.

C. Embeddings Generation

OpenAI's text-embedding-ada-002 model is a neural network that maps text to high-dimensional vectors. This means that it takes a piece of text and outputs a list of numbers, where each number represents a different feature of the text. For example, a number might represent the presence of a particular word in the text, or it might represent the overall sentiment of the text.

The text-embedding-ada-002 model is used in MediKnow to embed cancer-related documents into vectors. This process allows MediKnow to efficiently retrieve relevant documents based on user queries. For example, if a user submits a query about cancer symptoms, MediKnow can use the text-embedding-ada-002 model to find documents that are like the query, even if the documents do not contain the exact same words.

D. Vector Storage (Faiss)

Faiss, a specialized vector database, is employed for storing and querying the embedded cancer-related documents. It enhances the system's ability to retrieve the nearest neighbors to a given vector, facilitating efficient and accurate information retrieval.

E. Similarity Search: Query Processing with Faiss

Similarity search involves finding documents or data points that are most like a given query. In the "MediKnow", Faiss is used to perform similarity searches based on user queries. The following details the development of a function, get_similar_docs, to execute similarity searches and optionally include scoring for result relevance. Develop a function, get_similar_docs, to perform similarity searches using Faiss based on user queries.

Ensure the Faiss vector database (VDB) is properly initialized with the necessary configurations and parameters. Accept user queries within the get_similar_docs function. Utilize the OpenAIEmbeddings module with the "text-embedding-ada-002" model to embed the user query. This ensures consistency with the embeddings used for document chunks. Use Faiss to conduct a similarity search between the embedded user query and the embeddings of document chunks stored in the Faiss VDB. Associate scores with the retrieved documents to indicate their relevance to the user query. Retrieve the top-k documents based on similarity scores or similarity ranking from the Faiss VDB, providing users with the most relevant information.

F. Question Answering Chain

OpenAI's gpt-3.5-turbo model is a large language model that is trained on a massive dataset of text and code. It can generate human-quality text and understanding complex questions. The gpt-3.5-turbo model is used in MediKnow to generate answers to user queries. The model is given the retrieved relevant documents and the user's query, and it generates an answer that is based on the information in the documents and the context of the query. The gpt-3.5-turbo model can generate comprehensive and informative answers, even to complex questions.

a) Input Processing: Users input their queries in Malayalam related to cancer. Preprocessing: The input

queries are preprocessed to ensure uniformity, tokenization, and compatibility with the GPT-3.5 Turbo model.

b) Embeddings Generation: The preprocessed queries are converted into embeddings using the Turbo model underlying transformer architecture. The Turbo excels in understanding contextual information, capturing the meaning and nuances of the Malayalam language.

c) Context-Aware Question Generation: The system leverages the Turbo model to dynamically generate context-aware questions based on the input queries.Tthe Turbo model retains contextual information from the input, enhancing the relevance of the generated questions.

d) Context Retrieval from Similar Documents: The system utilizes Faiss for efficient vector storage and similarity searches to retrieve documents like the user's query. Documents containing relevant information are retrieved based on the embeddings generated by the Turbo model.

e) Question-Answering Chain: The context-aware questions and the retrieved documents are fed into the The Turbo model for question-answering tasks. Language Model Processing: It processes the input, understanding the context and generating detailed responses in Malayalam.

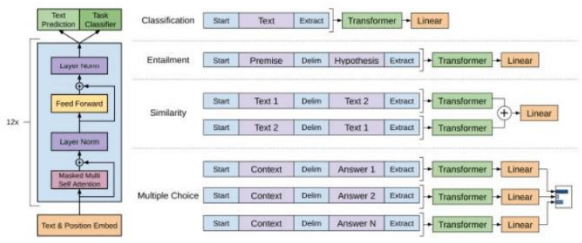
f) Answer Presentation: The system extracts answers generated by the Turbo model. For user transparency, both the generated answers and the context from the similar documents are presented

G. GPT 3.5 Model Architecture

GPT-3.5, Fig 3.1 [5] introduced in January 2022, represents a refined iteration of the GPT-3(Generative Pre-Trained Transformer) model, boasting three variations. One of its primary advancements is its enhanced capability to mitigate the generation of bad output. Featuring 12 decoder stacks with multi-head attention blocks.

This model incorporates 12 layers of decoder stacks and approximately 117 million parameters, trained on an extensive dataset exceeding 40GB of text [5]. Following the training phase, fine-tuning the model for specific tasks becomes essential, such as:

- Natural Language Inference
- Classification
- Question Answering
- Semantic Similarity



(left) Transformer architecture and training objectives used in this work. **(right)** Input transformations for fine-tuning on different tasks. We convert all structured inputs into token sequences to be processed by our pre-trained model, followed by a linear+softmax layer.

Fig 3.1 GPT 3.5 model

Unlike its predecessors, this model integrates human value-centric policies and operates with a reduced parameter count, with 1.3 billion parameters, a significant decrease compared to previous versions [5]. Dubbed as InstructGPT or GPT-3.5, this iteration utilizes reinforcement learning with human feedback (RLHF), a subfield of AI, to refine its performance. RLHF involves leveraging human input to enhance machine learning algorithms, addressing the limitations of traditional supervised and unsupervised learning methods Fig 3.2 [5].

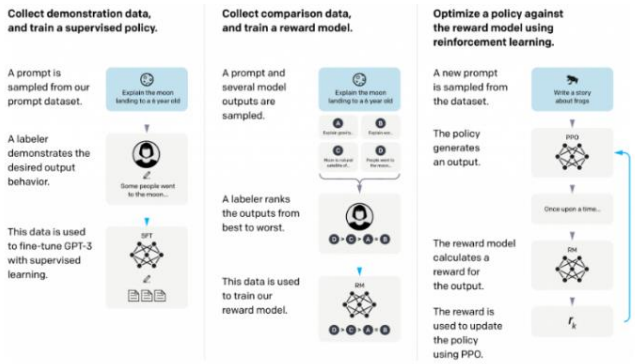


Fig 3.2 Reinforcement Learning with Human Feedback

The GPT-3.5 series encompasses models trained on a blend of text and code predating Q4 2021. Notable models within this series include

- code-davinci-002, which excels in code-completion tasks, and its derivative,
- text-davinci-002, an InstructGPT model.
- An improved version, text-davinci-003, further refines the capabilities introduced by its predecessors.

H. Answer Retrieval

In the "MediKnow", enhancing the answer retrieval process involves modifying the get_answer function to not only print the generated answers but also include information about the similar documents and their corresponding contexts. This modification aims to provide users with more transparency and a richer understanding of how the answers were generated. By modifying the get_answer function and enhancing the user interface, "MediKnow" improves the user experience by providing more comprehensive information about the generated answers and their context.

I. System Workflow

MediKnow is a Malayalam question answering system designed to provide comprehensive and accurate answers to cancer-related queries in Malayalam. The system in utilizes state-of-the-art generative models and natural language processing techniques to generate informative and contextually relevant responses.

a) Document Embedding: This component employs text-embedding-ada-002 model to embed cancer-related documents into a high-dimensional vector space. This process enables efficient retrieval of relevant information based on user queries. Each document is represented as a vector of numbers, where each number corresponds to a particular feature of the document. These vectors can then be used to quickly find documents that are like a given query.

b) Vector Storage: Faiss, a robust vector database, is employed for the efficient storage and management of the embedded documents. Faiss facilitates swift retrieval and processing of pertinent information during question answering. It is designed to handle large collections of vectors efficiently, optimizing operations like finding the nearest neighbors to a given vector, which is crucial in question answering tasks.

c) Question Answering: This component leverages gpt-3.5-turbo model to generate comprehensive and informative answers to user queries. The model draws upon the retrieved relevant documents and its knowledge base to provide contextually accurate responses. gpt-3.5-turbo is a large language model that is trained on a massive dataset of text and code. It can generate human-quality text and understanding complex questions.

When a user submits a question to MediKnow:

d) Query Processing: The user's query is first processed to extract keywords and identify relevant concepts. This helps to narrow down the scope of the search and ensures that the system retrieves the most relevant documents.

e) Document Retrieval: The processed query is then used to query the Faiss vector store. The system retrieves the documents that are most like the query, based on their vector representations.

f) Answer Generation: The retrieved documents are then passed to the gpt-3.5-turbo model, which generates an answer to the user's query. The model considers the context of the documents and the user's query to generate a comprehensive and informative response.

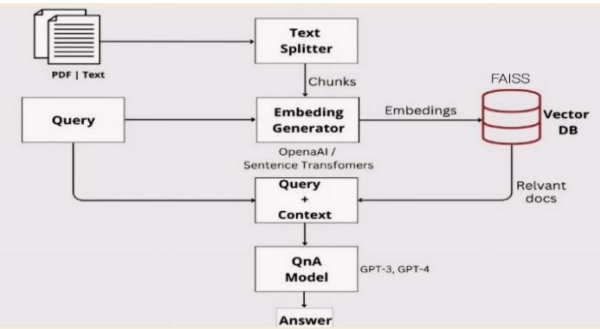


Fig 3.3 QAS Working Architecture

g) Answer Evaluation: The generated answer is then evaluated by the system to ensure that it is accurate, relevant, and complete. The system may also use additional resources, such as medical databases and dictionaries, to verify the correctness of the answer.

h) Answer Presentation: The final answer is presented to the user in a clear and concise manner. The system may also provide additional information, such as links to relevant resources, to help users further explore the topic.

IV. REQUIREMENTS

A) Libraries

Natural Language Toolkit, Werkzeug, OpenAI, Faiss, Langchain

B) Platforms

IDE: VSCODE, Google Colab

C) Software Configuration

Language: Python, Virtual Environment: Anaconda, Operating System: Windows 7 or later, Front End: Html, CSS, Backend: Flask

D) Hardware Configuration

- GPU: NVIDIA GeForce RTX 3080
- Architecture: Ampere
- CUDA Cores: 8704, Base Clock: 1440 MHz, Boost Clock: 1710 MHz
- Memory: 10 GB GDDR6X, Memory Interface: 320-bit, Memory Bandwidth: 760 GB/s
- Power Consumption: 320W, Interface: PCIe 4.0 x16

V. RESULT

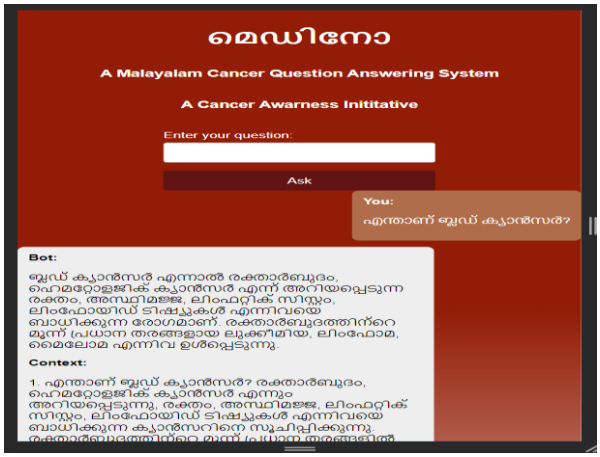


Fig 5.1 Web App Interface

MediKnow, a Malayalam question-answering system for cancer-related queries, represents a significant

advancement in leveraging natural language processing (NLP) technologies to address informational needs in the medical domain.

The paper successfully implemented a question-answering system capable of generating informative responses to cancer-related queries in Malayalam. Document loading, splitting, and embeddings generation were effectively implemented to handle Malayalam text data, providing a foundation for contextual understanding. Faiss's similarity search functionality enhanced the system's ability to retrieve relevant documents based on user queries, contributing to a more dynamic and interactive user experience. The paper successfully integrated OpenAI's language models, particularly GPT-3.5 Turbo, to generate context-aware answers, showcasing the versatility of large-scale language models in medical information retrieval. The user interface improvements, including the presentation of context alongside answers, contribute to a more user-centric experience.

VI. CONCLUSION

- Context-Aware Question Answering: The paper has effectively implemented a system capable of generating context-aware answers to cancer-related questions in the Malayalam language. This is a significant accomplishment in leveraging advanced language models for medical information retrieval.
- Document Processing and Embeddings: The implementation of document loading, splitting, and embeddings generation processes demonstrates the paper's ability to handle Malayalam text data efficiently, laying the groundwork for nuanced contextual understanding.
- Language Model Integration: Successfully incorporating OpenAI's language models, especially GPT-3.5 Turbo, showcases the versatility of large-scale language models in understanding and generating content in the medical domain.
- User Interface Enhancements: The improvements made to the user interface, particularly in presenting context alongside answers, enhance user understanding and contribute to a more user-centric experience.

ACKNOWLEDGMENTS

The author(s) would like to thank Mr. Dhanunath R, Assistant Professor, Department of Computer Science and Engineering, Sree Buddha College of Engineering, Pattoor for his constant support and mentorship.

REFERENCES

[1] Question Answering over Linked Data with GPT-3

[2] Jurafsky D and Martin J H , Speech and Language Processing An Introduction to Natural Language Processing, Computational Linguistics and Speech Recognition , Pearson Education Series 2002

[3] Generating answers for a question-answering systems, January 2012, V.-M. Pho

[4] JamesAllen, Natural Language Understanding, Benjamin/Cummings Publishing Company,1995

[5] Ahmed Mandour n.d., *OpenGenus IQ: Computing Expertise & Legacy*—GPT-3.5 model architecture, 02 November 2023, <https://iq.opengenus.org/gpt-3-5-model/>

[6] Sara Renjit and Sumam Idicula "Natural language inference for Malayalam language using language agnostic sentence representation", May 2021

[7] Antony P.J, Santhanu P Mohan, Soman K.P "SVM Based Part of Speech Tagger for Malayalam", June 2019

[8] Question Answering over Linked Data with GPT-3 - Bruno Faria, Dylan Perdigão, Hugo Gonçalo Oliveira, August 2023

[9] Bibin P, Babu Anto "Malayalam Questions Classification in Question Answering Systems using Support Vector Machine", Jan 2019