Medical Chatbot for Cancer Information Retrieval

By Group 8

Problem Statement

Challenge

Patients face significant difficulties in accessing reliable, timely cancer-specific information online.

Consequence

This leads to widespread misinformation, increased anxiety for patients, and places an unsustainable burden on healthcare professionals who are already stretched thin by routine inquiries.

Project Objectives

1. **Develop Robust Text Preprocessing:** Create a sophisticated pipeline to clean and normalize diverse medical text data from the MedQuAD dataset, ensuring optimal input for our NLP models.
2. **Implement Effective Hybrid Retrieval:** Design and build a powerful system that intelligently combines keyword-based matching (TF-IDF) with advanced semantic understanding (BioBERT embeddings) to accurately retrieve relevant Q&A pairs.
3. **Evaluate Model Performance Rigorously:** Quantitatively assess the retrieval model's accuracy, efficiency, and relevance using comprehensive metrics and diverse test cases to meet predefined performance goals.
4. **Prepare for API Integration:** Structure the core NLP model's components and functionalities to be easily deployable as a scalable backend service, ready for integration into various chatbot applications.
5. **Enhance User Experience:** Contribute to a system that provides immediate, trustworthy, and highly relevant information, ultimately improving how individuals access and understand complex cancer-related knowledge.

Key Stakeholders:

- **Patients/General Public:** Will gain quick, reliable access to validated cancer information, reducing reliance on potentially inaccurate online sources.
- **Healthcare Providers/Hospitals:** Can integrate this solution to streamline patient support services, reduce the volume of routine inquiries, and free up staff for more critical tasks.
- **Medical Researchers/Educators:** Can utilize the structured Q&A data for analytical purposes, educational content development, and to identify common information gaps.

Data Understanding

Dataset: MedQuAD (Medical Question-Answer Dataset)

Source: Curated from trusted medical websites (e.g., Cancer.gov).

Format: XML files containing structured Question-Answer pairs.

Data Loading & Overview:

Custom Loader: Developed CancerQALoader to parse XML efficiently and consolidate data.

Initial Size: 729 Q&A pairs, 3 columns (question, answer, source).

Unique Sources: 116 distinct source files, indicating broad coverage.

Key Features:

question: User query component; basis for retrieval.

answer: Authoritative response to be retrieved.

source: Provides metadata and reinforces trustworthiness.

Data Cleaning and Preprocessing

The raw dataset contained redundant or near-duplicate question-answer pairs, which could bias our model's training and lead to repetitive or less accurate retrieval.

Action: Removed duplicate questions, retaining the first occurrence, to ensure data integrity and prevent skewing model understanding.

Result: Reduced dataset size from 729 to 683 unique Q&A pairs.

Data Cleaning and Preprocessing

We developed a multi-stage, custom clean_text pipeline tailored to the specifics of medical language:

- Medical-Specific Stopwords: Beyond general English stopwords, we utilized a curated list of medical-specific stopwords (e.g., "patient," "doctor," "medical") to remove terms that are common but carry little discriminatory power in our domain.
- Boilerplate Removal: We systematically eliminated common boilerplate phrases (such as "Key Points" or source indicators) and extraneous HTML tags that were present in the raw data. This step ensures that only relevant informational content is processed.
- Case Normalization: All text was converted to lowercase to treat variations in capitalization uniformly. Crucially, we preserved hyphens within medical terms (e.g., "T-cell," "B-cell") to maintain their specific meaning.
- Tokenization & Lemmatization: Text was broken down into individual words (tokenization), and then words were reduced to their base or dictionary form (lemmatization) (e.g., "therapies" to "therapy," "diseases" to "disease"). This step helps in matching variations of words.

Modelling

Before our chatbot can find answers, it needs to "understand" questions and answers. This step is about converting human language into a numerical format that computers can process and compare. We do this in two main ways: for keyword matching and for understanding meaning.

1. Keyword Matching: TF-IDF (Term Frequency-Inverse Document Frequency):

- We used a tool called TfidfVectorizer.
- It analyzes all our cleaned questions, identifies important keywords, and creates a unique numerical "fingerprint" for each question based on these keywords.

Modelling

2. Understanding Meaning: BioBERT Embeddings

- This is where our chatbot gets its "smart" understanding. BioBERT is a powerful AI model specifically trained on vast amounts of medical text. It doesn't just look at words; it understands their context and meaning. Think of it like giving each question a "meaning fingerprint.“
- By using both TF-IDF and BioBERT, we give our chatbot two powerful ways to "listen" to a user's question: one for precise keywords and another for deeper meaning. These numerical representations are the essential building blocks that allow our hybrid search system to find the most relevant answers.

Hybrid Search System

- We developed a hybrid search function that combines the strengths of both keyword-based and semantic understanding methods.
- **Components of the Hybrid Search:**
- **TF-IDF (Term Frequency-Inverse Document Frequency):**
    - **Function:** Measures the importance of a word in a document relative to a collection of documents.
- **BioBERT Semantic Embeddings**:
    - **Function**: A specialized BERT-based model that converts text into dense numerical vectors (embeddings).

Model Evaluation

We employed a structured evaluation framework that allowed us to quantitatively measure the model's ability to retrieve accurate and relevant information.

A carefully curated validation_data set was created.

Each test case included:

- query: The user's input question.
- expected: A list of keywords anticipated in a relevant answer.
- disease: An optional filter for disease-specific queries.
- min_similarity: A minimum confidence threshold tailored per query.

Model Evaluation

The evaluate_model Function: This custom function systematically runs each test case through the hybrid_search function.

The function provides granular feedback for every test case (detailing the original query, found keywords, the best match's question and partial answer, its similarity, and a clear pass/fail status).

Critically, it also generates an overall summary, highlighting total tests, passed, failed, and the cumulative accuracy, alongside detailed reasons for each failure.

Key Findings from Evaluation & Tuning

Initial Performance Insight: Our initial evaluation on the expanded dataset provided a baseline accuracy of 71.43%, confirming the potential of our approach but also highlighting clear opportunities for refinement.

Iterative Optimization Success: This rigorous evaluation process was the cornerstone of our iterative optimization. It directly guided our hyperparameter tuning efforts for tfidf_weight, semantic_weight, and the internal threshold

Optimization Results:Through rigorous tuning, we determined the optimal configuration for our hybrid_search function:

- Semantic Weight: 1.00
- TF-IDF Weight: 0.00
- Internal Threshold: 0.30
- **Final Achieved Accuracy: 85.71%**
- **Result:** After applying these optimized parameters (TF-IDF=0.00, Semantic=1.00, Threshold=0.30), our chatbot achieved a final accuracy of **85.71%** on our comprehensive validation set.