CSE 4/535-Introduction to Information Retrieval - Fall 2024

Project - 3

The Q&A Wikipedia Chatbot

Team Member Details:

1. Name: Karan Ramchandani

UB Person: 50610533

2. Name: Vikranth Bandaru

UB Person: 50607347

Table of Contents

1. Introduction

The Q&A Wikipedia chatbot is a web-based application that uses sophisticated Natural Language Processing (NLP) techniques to give users both lighthearted responses and in-depth answers to topic-based requests.

The chatbot:

- Allows chit-chat, resulting in a human-like and conversational experience.
- Retrieves and condenses Wikipedia content for queries pertaining to ten predetermined subjects.
- Features an analytics dashboard and an easy-to-use web interface for visualizing chatbot performance and inquiry patterns.

This project expands on prior understanding of:

Project 1: Developed a wiki scraper and used Solr to index documents.
Project 2: Created unique scoring and document retrieval systems.

For a smooth Q/A experience, all of the project's components were combined into a single Retrieval-Augmented Generation (RAG) pipeline.

2. Methodology

2.1 Wiki Scraper

Objective: For each of the ten topics, at least 5000 distinct Wikipedia articles were scraped.
Tools Used:

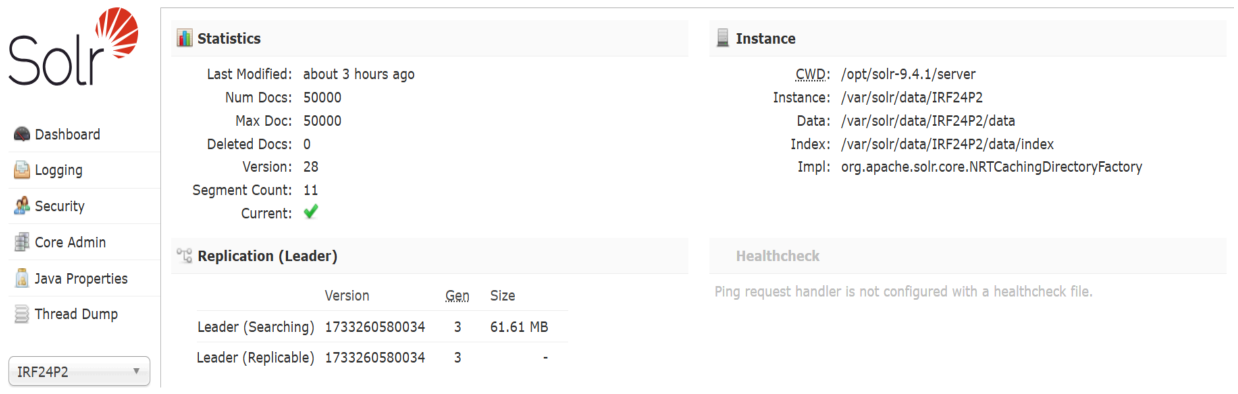
- ScraperWiki is the tool used for scraping.
- Hashing is used in deduplication to guarantee unique records.

Topics Scraped:

- Health
- Environment
- Technology
- Economy
- Entertainment
- Sports
- Politics
- Education
- Travel
- Food

These are among the topics that were scraped.

- Made sure that more than 50,000 documents were distributed evenly among the subjects.



Statistics

```
   Last Modified:  about 3 hours ago
        Num Docs:  50000
         Max Doc:  50000
    Deleted Docs:  0
         Version:  28
   Segment Count:  11
         Current:  ✓
```

Instance

```
      CWD:  /opt/solr-9.4.1/server
 Instance:  /var/solr/data/IRF24P2
     Data:  /var/solr/data/IRF24P2/data
    Index:  /var/solr/data/IRF24P2/data/index
     Impl:  org.apache.solr.core.NRTCachingDirectoryFactory
```

Replication (Leader)

	Version	Gen	Size
Leader (Searching)	1733260580034	3	61.61 MB
Leader (Replicable)	1733260580034	3	-

Healthcheck

Ping request handler is not configured with a healthcheck file.

2.2 Chit-Chat

- **Model**: Implemented chit-chat using **OpenAI GPT-3.5 Turbo** with prompts for a conversational assistant.
- **Prompt**:

  To determine the conversational style of the assistant, OpenAI's GPT model is given a few-shot example as the prompt:

```
# Few-shot prompt to guide the assistant's conversational style
prompt = f"""
You are a friendly, conversational assistant. Respond to casual, informal questions or comments in a natural and engaging manner

Examples:
- User: "What's your favorite food?"
  Assistant: "I don't eat, but if I could, I'd love to try pizza!"
- User: "How's the weather there?"
  Assistant: "It’s always perfect here in the digital world. How about you?"
- User: "Tell me a joke."
  Assistant: "Why don't skeletons fight each other? Because they don’t have the guts!"

Now respond to this query:
User: "{query}"
Assistant:
"""
```

The assistant is designed to be amiable and interesting, and it will naturally answer questions that are informal or casual.

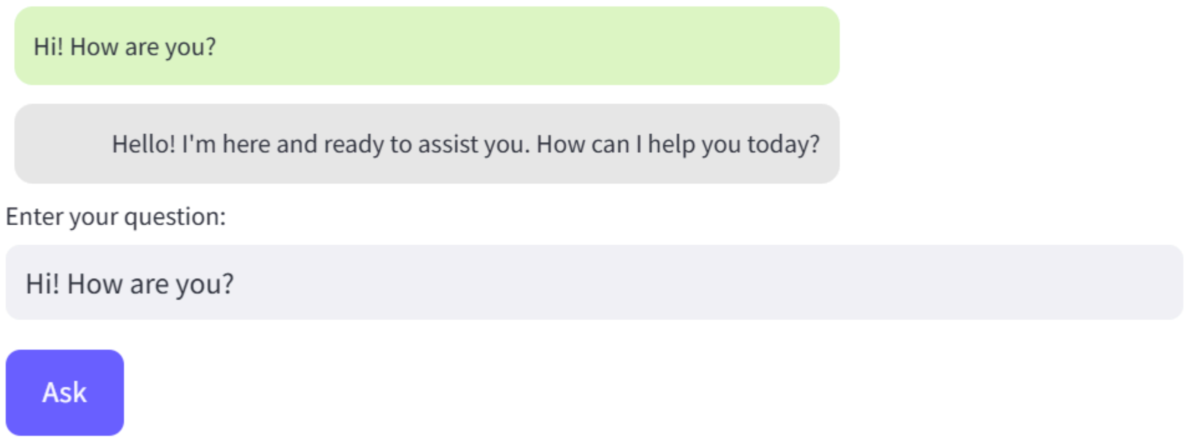
The prompt's examples show:

- How the helper ought to react when asked about personal preferences, such as "What's your favorite food?"
- Informal questions, such as the weather.
- Humorous requests, such as "Tell me a joke."

With the help of this leading suggestion, the chatbot is guaranteed to keep up a conversational tone that is human-like throughout small talk.

The model ensures a flow of natural dialogue without being rule-based.

An Example of an Interface:


Hi! How are you?
Hello! I'm here and ready to assist you. How can I help you today?
Enter your question:
Hi! How are you?
Ask

User Input: The user types a lighthearted question, such as "Hey! "How are you?"
Assistant Reaction: The chatbot responds with a kind and amiable salutation, such as "Hello! “I'm here and ready to assist you. How can I help you today?"
Design Components:
User Message: For easy differentiation, it is shown in a green bubble.
The assistant message is shown in a grey bubble and is designed to be easily readable.
The "Ask" button and input box allow users to interact.

2.3 Multi Topic Classification(Two level classification)

- **Objective:** Classify queries into one or more predefined topics.
- **Method:**
    - A few-shot learning approach using OpenAI GPT.
    - Examples were provided in the prompt to guide the model:

Prompt for Chit-chat/Topic Related Query Classification:

```
def classify_query(query):
    """
    Classify the query as either 'chit-chat' or 'topic-related'.
    Uses a few-shot learning prompt for classification.
    """
    prompt = """
    Classify the following queries as either 'chit-chat' or 'topic-related'. Respond with exactly one of the two labels.

    Examples:
    1. "How's the weather today?" => chit-chat
    2. "Can you tell me more about deep learning?" => topic-related
    3. "What's your favorite movie?" => chit-chat
    4. "Explain the benefits of federated learning." => topic-related

    Query: "{}"
    respond only either chit-chat ot topic-related
    """.format(query)
```

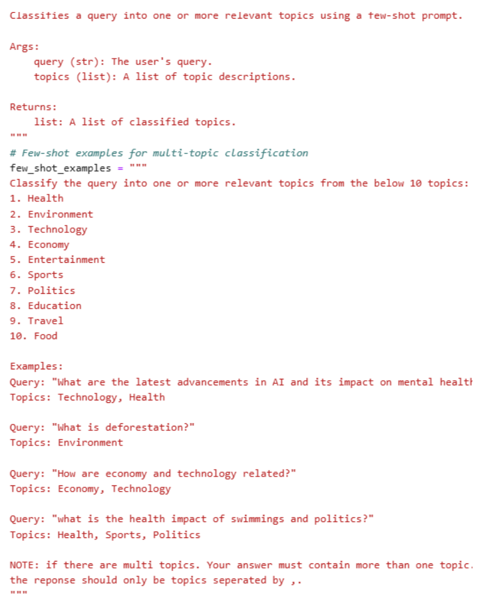
A prompt for categorizing searches as "topic-related" or "chit-chat" is depicted in the above image.

Purpose: Determine whether a query is informal (like "How's the weather?") or specifically aims to provide knowledge (like "Explain deep learning").

Method: The model's reaction is guided by labeled samples provided by a few-shot prompt.

Result: Directs topic-related questions to the Wiki Q/A system for pertinent responses, and informal questions to the chit-chat module.

Prompt of Topic Classification:



```
Classifies a query into one or more relevant topics using a few-shot prompt.

Args:
    query (str): The user's query.
    topics (list): A list of topic descriptions.

Returns:
    list: A list of classified topics.
"""
# Few-shot examples for multi-topic classification
few_shot_examples = """
Classify the query into one or more relevant topics from the below 10 topics:
1. Health
2. Environment
3. Technology
4. Economy
5. Entertainment
6. Sports
7. Politics
8. Education
9. Travel
10. Food

Examples:
Query: "What are the latest advancements in AI and its impact on mental healtl
Topics: Technology, Health

Query: "What is deforestation?"
Topics: Environment

Query: "How are economy and technology related?"
Topics: Economy, Technology

Query: "what is the health impact of swimmings and politics?"
Topics: Health, Sports, Politics

NOTE: if there are multi topics. Your answer must contain more than one topic.
the reponse should only be topics seperated by ,.
"""
```

This prompt helps classify subjects by identifying which of the ten predetermined topics (such as technology or health) a user question pertains to.

Purpose: Sorting questions into one or more pertinent categories is the goal.

Method: Sample questions and related subjects are included in a few-shot example-based prompt.

Example: "What is deforestation?" is the question. Environment is one of the topics.

Result: Directs the chatbot to concentrate its answers on the designated subjects for increased precision and pertinence.

For topic-based queries, the model filters relevant topics efficiently for downstream processes.

2.4 Wiki Q/A Bot (Summarization using LLM)

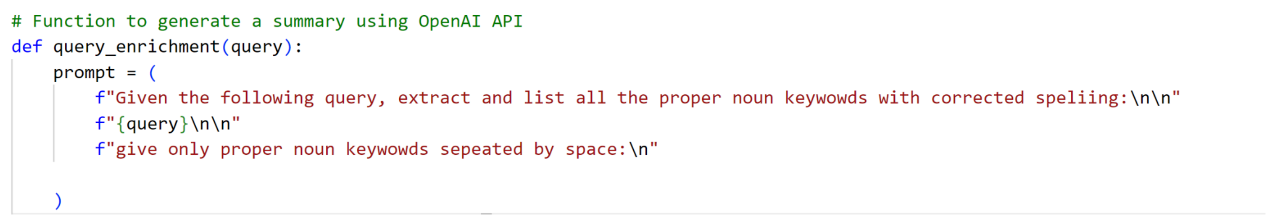
The objective is to extract pertinent articles from Wikipedia and condense them into summary

Using the GPT3.5 model.

Steps:

2.4.1 Keyword Extraction and Query Expansion for correct retrieval

OpenAI GPT was used to extract keywords correctly:



```
# Function to generate a summary using OpenAI API
def query_enrichment(query):
    prompt = (
        f"Given the following query, extract and list all the proper noun keywowds with corrected speliing:\n\n"
        f"{query}\n\n"
        f"give only proper noun keywowds sepeated by space:\n"

    )
```

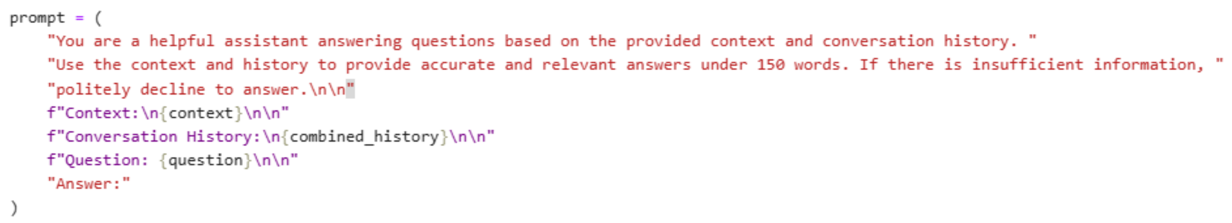
Guarantees accurate, enhanced searches for document retrieval.

2.4.2 Document Retrieval

- The backend for obtaining pertinent documents based on enriched queries was Apache Solr.
- Further filtered results according to previously classed subjects.

2.4.3 Summarization

Using the recovered documents, OpenAI GPT produced summarized responses:



```
prompt = (
    "You are a helpful assistant answering questions based on the provided context and conversation history. "
    "Use the context and history to provide accurate and relevant answers under 150 words. If there is insufficient information, "
    "politely decline to answer.\n\n"
    f"Context:\n{context}\n\n"
    f"Conversation History:\n{combined_history}\n\n"
    f"Question: {question}\n\n"
    "Answer:"
)
```

This section describes how OpenAI GPT was used to produce summarization.

Purpose: Produce succinct, precise responses by integrating:

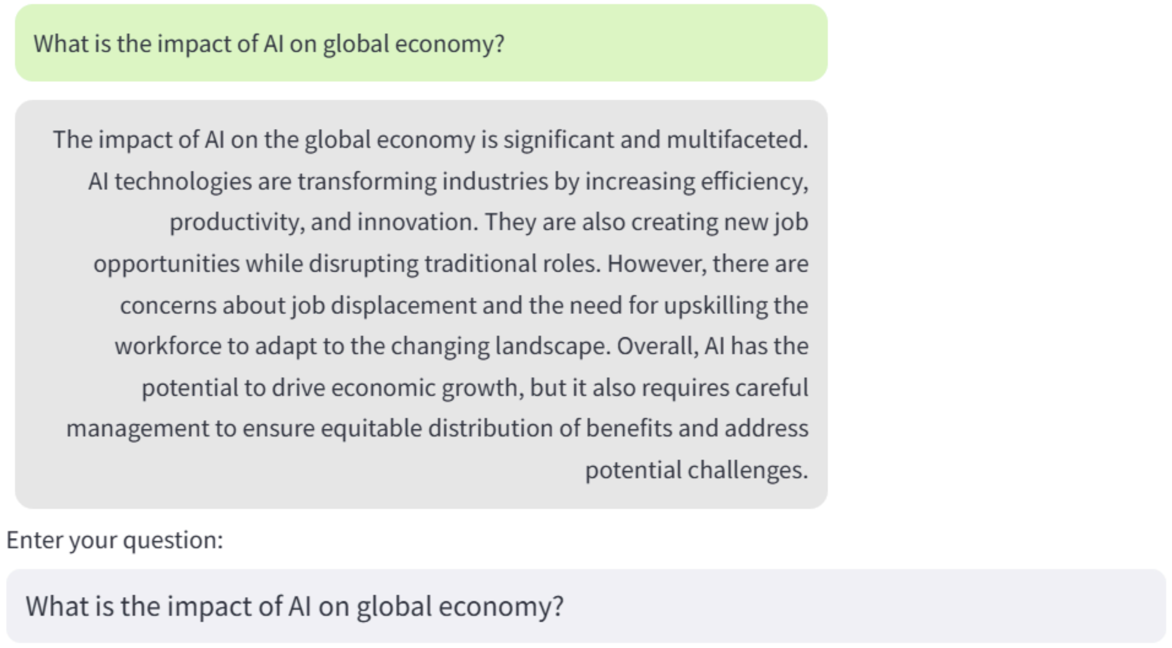
- **Context:** Documents that were relevantly obtained.

- **Conversation History:** Previous exchanges to maintain continuity.

Prompt: Tells the model to respond with no more than 150 words or to respectfully decline if there is not enough information.

Result: Assures that answers are pertinent to the user's question, coherent, and cognizant of context.

Example of summarization on Topic related query:

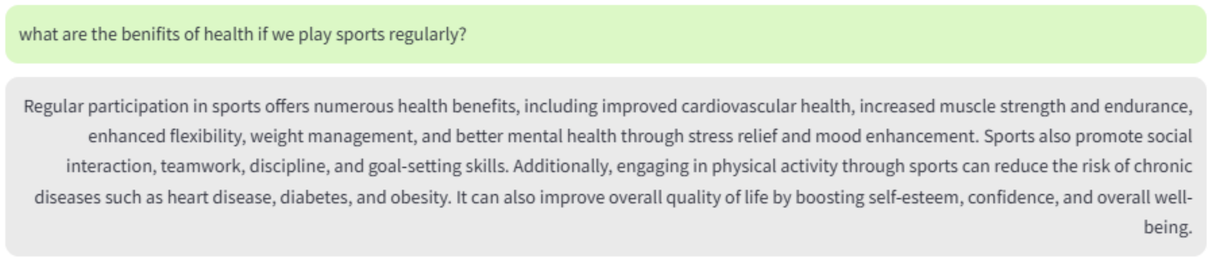


What is the impact of AI on global economy?

The impact of AI on the global economy is significant and multifaceted. AI technologies are transforming industries by increasing efficiency, productivity, and innovation. They are also creating new job opportunities while disrupting traditional roles. However, there are concerns about job displacement and the need for upskilling the workforce to adapt to the changing landscape. Overall, AI has the potential to drive economic growth, but it also requires careful management to ensure equitable distribution of benefits and address potential challenges.

Enter your question:

What is the impact of AI on global economy?



what are the benifits of health if we play sports regularly?

Regular participation in sports offers numerous health benefits, including improved cardiovascular health, increased muscle strength and endurance, enhanced flexibility, weight management, and better mental health through stress relief and mood enhancement. Sports also promote social interaction, teamwork, discipline, and goal-setting skills. Additionally, engaging in physical activity through sports can reduce the risk of chronic diseases such as heart disease, diabetes, and obesity. It can also improve overall quality of life by boosting self-esteem, confidence, and overall well-being.

This example shows how to use OpenAI GPT and retrieved documents to summarize results for topic-related questions.

Query: "What is the impact of AI on the global economy?" is one example of a specific question that users pose.

Response: Using the pages that were retrieved, the model creates a succinct, educational synopsis that covers the main points of the question.

Result: Offers concise, well-organized responses, such as stressing the advantages of regular sports engagement for health or AI's contribution to efficiency, job creation, and economic progress.

This guarantees context-aware, pertinent answers to user inquiries.

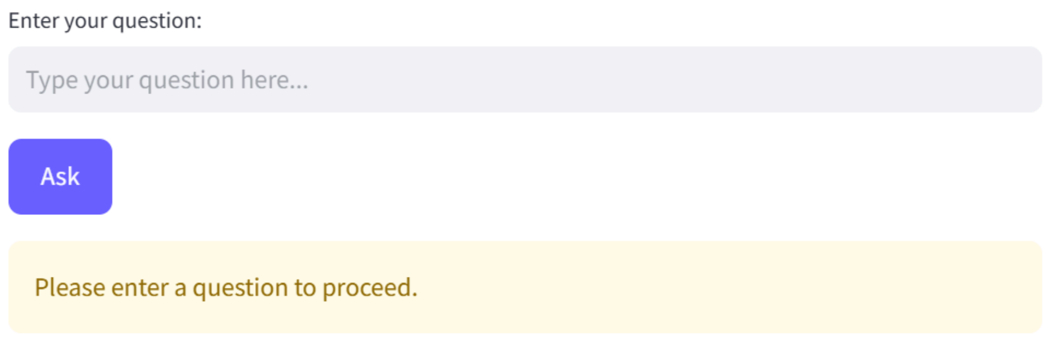
2.5 Exceptional Handling

Why is it needed?

- When the chatbot is unable to process a query, stop it from crashing.
- Make sure your fallback responses are elegant.

Implemented Scenarios:

- **Not classified Questions:** The bot will reply, "I'm sorry, I don't have enough information on that," if a question doesn't relate to any topic.
- **Empty or Invalid Responses:** Addressed situations in which summaries are lacking or when the Solr retrieval yields no results.
- **Empty Queries:** When a user ask the chatbot without entering anything then it gives out the output as "Please enter a question to proceed"


Enter your question:

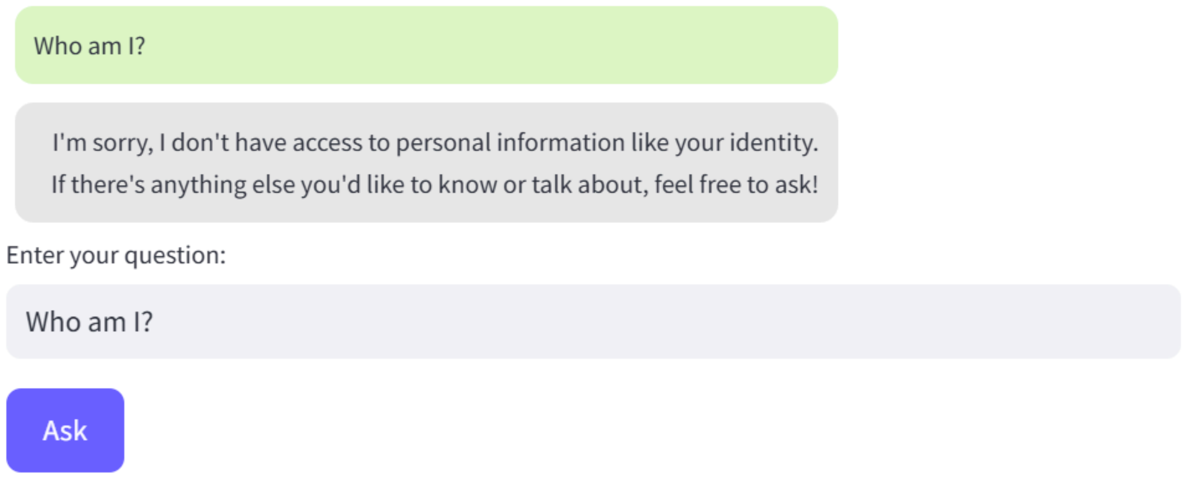
Type your question here...

Ask

Please enter a question to proceed.

- **Handling Personal or Sensitive Queries:**

  This illustration shows how the chatbot can respond to private questions. In a courteous and businesslike manner, the bot responds to the question "Who am I?" by explaining its restrictions, including the fact that it is unable to access personal data.


Who am I?

I'm sorry, I don't have access to personal information like your identity. If there's anything else you'd like to know or talk about, feel free to ask!

2.6 Visualization and UI

- **Web Application**:

Built using **Streamlit**.

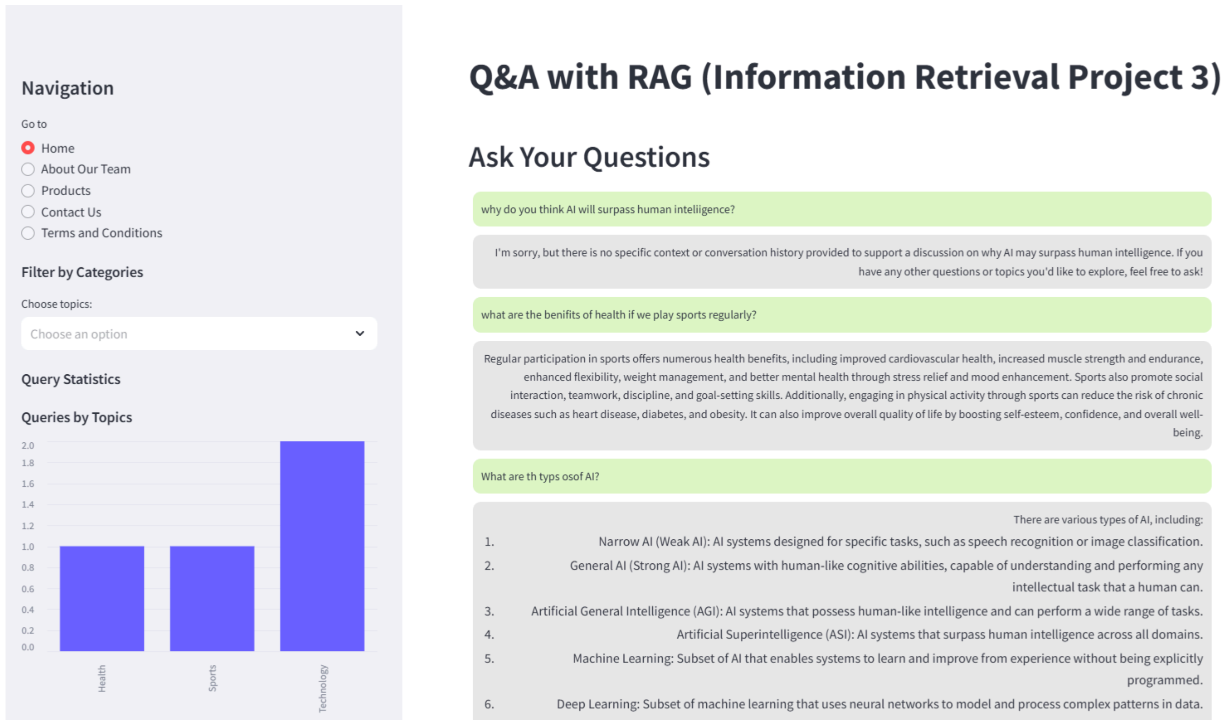
With the help of the open-source Python module Streamlit, you can rapidly and with little code construct dynamic, data-driven web apps. Because it allows users to observe and interact with data in real time, it is especially helpful when developing machine learning and data science applications.

Important Streamlit Features:

- ○ **Simple to Use:** Just a few lines of code are needed to turn Python scripts into interactive web applications.
- ○ **Instant Feedback:** Streamlit is perfect for prototyping because it refreshes the application in real time as you make changes to the code.
- ○ **No Frontend Knowledge Required:** HTML, CSS, and JavaScript are not required, in contrast to standard web development. Python is used to write everything.
- ○ **Data Visualizations:** It makes it simple to incorporate plots and graphs by supporting well-known visualization libraries (such as Matplotlib, Plotly, and Altair).

- Hosted on **Google Cloud Platform (GCP)** for accessibility.

- **Chat Interface**:
    - Displayed user and assistant messages with styled bubbles for a clean UI.



Navigation

Go to

- [✓ Home]
- About Our Team
- Products
- Contact Us
- Terms and Conditions

Filter by Categories

Choose topics:

Choose an option

Query Statistics

Queries by Topics

Q&A with RAG (Information Retrieval Project 3)

Ask Your Questions

why do you think AI will surpass human inteliigence?

I'm sorry, but there is no specific context or conversation history provided to support a discussion on why AI may surpass human intelligence. If you have any other questions or topics you'd like to explore, feel free to ask!

what are the benifits of health if we play sports regularly?

Regular participation in sports offers numerous health benefits, including improved cardiovascular health, increased muscle strength and endurance, enhanced flexibility, weight management, and better mental health through stress relief and mood enhancement. Sports also promote social interaction, teamwork, discipline, and goal-setting skills. Additionally, engaging in physical activity through sports can reduce the risk of chronic diseases such as heart disease, diabetes, and obesity. It can also improve overall quality of life by boosting self-esteem, confidence, and overall well-being.

What are th typs osof AI?

There are various types of AI, including:

1. Narrow AI (Weak AI): AI systems designed for specific tasks, such as speech recognition or image classification.
2. General AI (Strong AI): AI systems with human-like cognitive abilities, capable of understanding and performing any intellectual task that a human can.
3. Artificial General Intelligence (AGI): AI systems that possess human-like intelligence and can perform a wide range of tasks.
4. Artificial Superintelligence (ASI): AI systems that surpass human intelligence across all domains.
5. Machine Learning: Subset of AI that enables systems to learn and improve from experience without being explicitly programmed.
6. Deep Learning: Subset of machine learning that uses neural networks to model and process complex patterns in data.

The Q&A chatbot's Chat Interface is displayed in this picture.

Important Features

Navigation panel: It offers the ability to view query statistics (such as queries by topic), filter topics, and browse categories.

Styled Chat Bubbles: For a clear and interesting user interface, user inquiries and assistant answers are shown in visually separate bubbles.

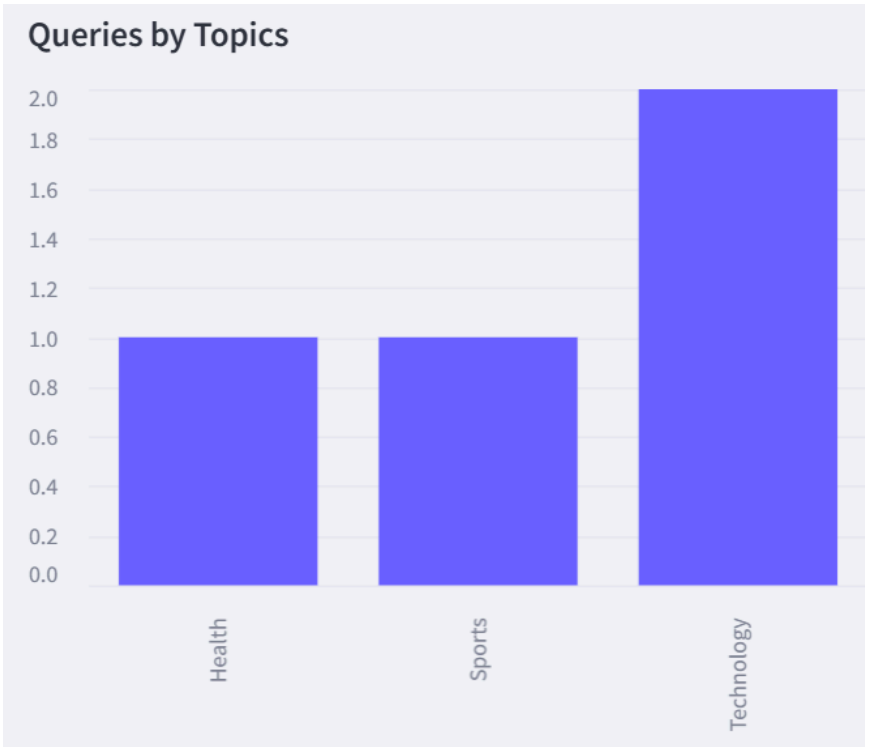
Real-Time Responses: Using information that has been retrieved, the assistant responds to queries from users in real time with pertinent, context-aware responses.

- **Analytics Dashboard**:

    Visualizations Included:

    - **Queries by Topic:** Bar chart showing the distribution of queries across topics.
    - **Trends Over Time:** Line chart displaying the number of terms per query over time.
    - **Top Queries:** Table of the most frequent user queries.

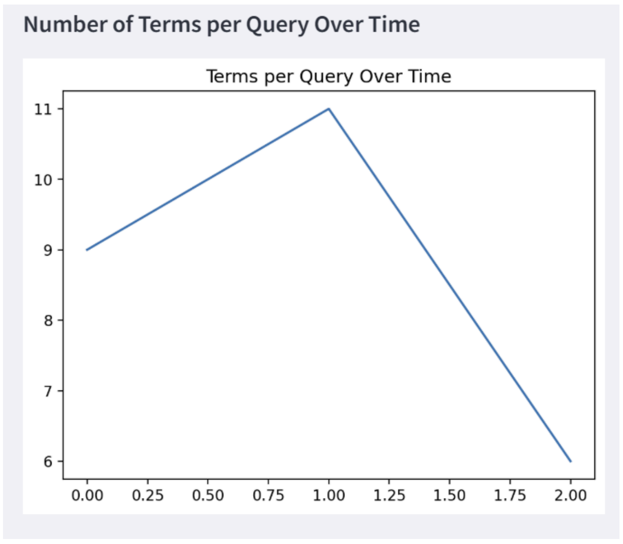
- **Query Grouping:** Groups similar queries based on length or content.



Queries by Topics

Purpose: The distribution of user searches across predetermined subjects is displayed for this purpose.

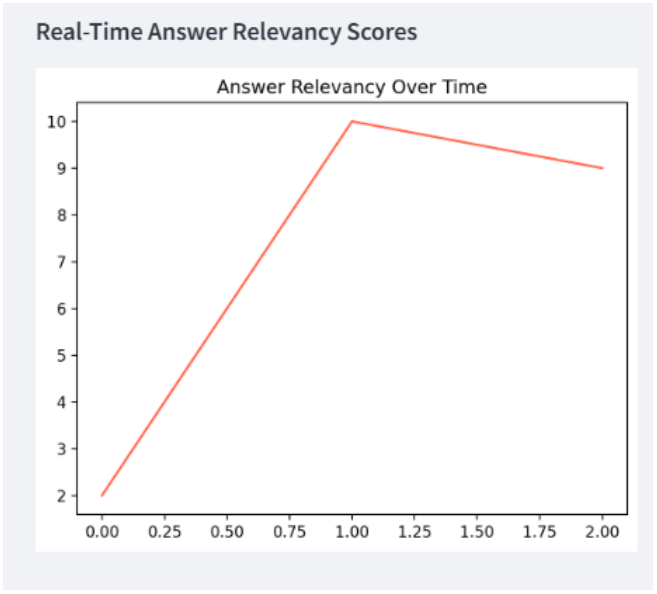
Insights: User interest statistics show that technology has the most questions, followed by sports and health.



Number of Terms per Query Over Time

Goal: Monitors the typical quantity of phrases in user inquiries over time.

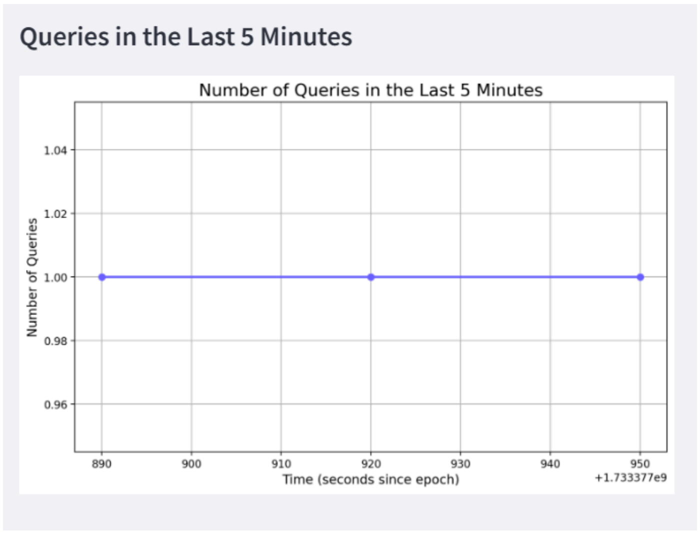
Insights: Indicates that users' query complexity varies, with more complicated queries (higher term count) at first and simpler ones subsequently.



Real-Time Answer Relevancy Scores:

Purpose: Evaluates over time how pertinent the chatbot's answers are to user inquiries.

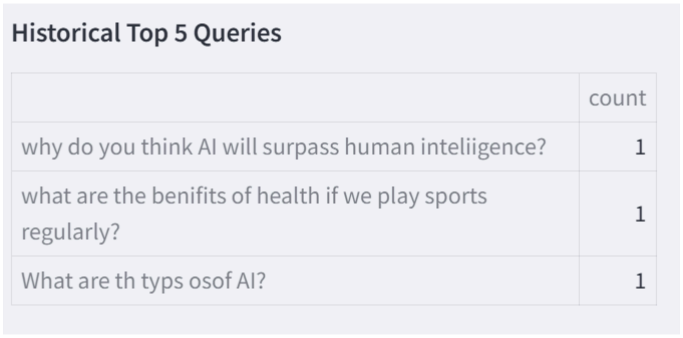
Insights: While modest drops point to places for growth, strong relevance scores show that the chatbot gives reliable answers.



Queries in the Last 5 Minutes

Purpose: Shows query activity within the previous five minutes.

Insights: Shows how frequently users engage, which is useful for tracking chatbot usage in real time.



Historical Top 5 Queries

	count
why do you think AI will surpass human inteliigence?	1
what are the benifits of health if we play sports regularly?	1
What are th typs osof AI?	1

Historical Top 5 Queries

Purpose: Provides a list of the most common questions people ask.

Insights: Draws attention to topics of shared interest, like the advantages of participating in sports and AI's potential to outsmart humans.

These visualizations aid in the analysis of chatbot performance, user behavior, and possible areas for development.

3. Work Breakdown by Teammates

Since this project was carried out on an individual basis:

- **Wiki Scraping**: Vikranth
- **Chit-chat Model**: Karan
- **Topic Classification**: Karan
- **Wiki Q/A Bot**: Karan
- **Visualization/UI**: Vikranth
- **Report:** Karan and Vikrant
- **Video:** Karan

4. Conclusion

4.1 Summary of the Achievements

Chit-chat and information retrieval are skillfully combined by the chatbot to create a smooth user experience. Among the notable achievements are:

- Precise classification of topics and retrieval of documents.
- Real-time query analytics on an intuitive online interface.
- Strong exception management for erroneous inquiries or answers.

4.2 Challenges handled

- Managing huge datasets—more than 50,000 documents.
- Ensuring distinct documents for every subject.
- Enhancing summarization and retrieval for performance in real time.

4.3 Future Work and Improvements

- Adjusting small talk answers to make them more unique.
- Extending subjects and adding multimedia responses (such as pictures and videos).
- Enhancing query enrichment and speeding up summarization.

5. References and Resources

5.1 Libraries and Tools Used

The libraries and tools used in the project are listed below, along with an explanation of their functions:

Libraries for Python

Streamlit: The interactive online application was developed using Streamlit, which offers a straightforward method for creating dashboards and user interface elements.

OpenAI GPT: Used for subject classification, query enrichment, summarization, and chit-chat response generation (via the `openai` library).

Pysolr: Used to communicate with the Apache Solr backend to get documents.

Matplotlib: Used in the analytics dashboard to create visualizations like bar graphs and line charts.

Pandas: Used to handle query information, manipulate data, and produce analytics insights.

Traceback: In order to prevent the application from crashing when errors occur, traceback is used for debugging and exception handling.

Tools for the Backend

Apache Solr: The Apache Solr search engine is used to index and retrieve documents that have been scraped from Wikipedia.

ChromaDB: A vector database used for retrieval-augmented generation (RAG) that enhances context and embeds queries.

Tools for Web Hosting

Google Cloud Platform (GCP):The Streamlit web application is hosted on Google Cloud Platform (GCP), which makes it available to the general public.

Additional Tools for Development

Python 3.8+: The complete implementation is written in this programming language.

Git: A version control solution for teamwork and code change management.

5.2 External Resources and Documentation

A list of outside sources and documents consulted throughout the process is provided below:

Documentation

OpenAI API documentation: This is used to comprehend and incorporate GPT-3.5 for tasks involving summarization, topic classification, and small talk.

https://platform.openai.com/docs

Streamlit Documentation: Offers instructions for creating the application's user interface and putting interactive widgets in place.

https://docs.streamlit.io

Apache Solr Documentation: Assists in setting up Solr for indexing and retrieving documents.

https://solr.apache.org

ChromaDB Documentation: Used to configure and communicate with the RAG vector database.

https://www.trychroma.com

Datasets

Wikipedia: The main resource for extracting more than 50,000 documents on ten predetermined subjects.

Chit-chat Dataset: Used as a guide when creating chit-chat questions.

http://github.com/BYU-PCCL/chitchat-dataset

Blogs and Online Tutorials

Documentation for ChatterBots: For preliminary concepts on chit-chat platforms.

https://chatterbot.readthedocs.io/en/stable

Python Exception Handling: This is used to create a comprehensive application error management system.

https://docs.python.org/3/tutorial/errors.html

Best Practices for Visualization: Used to create charts that are understandable and instructive.

https://matplotlib.org/stable/gallery/index.html

These materials played a crucial role in making sure the chatbot was developed and implemented successfully.