CSE 535: Information Retrieval
Project4: Multi-topic Information Retrieval Chatbot

Team: WebSpiders

Name: Mohit Sai Aravind Nunna
UB ID: mnunna (50468322)

Name: Yasmeen Mohammed
UB ID: ymohamme (50465533)

Name: Saad Ahmed
UB ID: saadahm2 (50483867)

Name: Abhigna Sree Tumati
UB ID: abhignas (50470999)

1. Introduction:

A chatbot is software that simulates human-like conversations with users via text messages on chat. Its key task is to help users by providing answers to their questions.
In the current project, we have built one such chatbot that interacts and enables communication between itself and the user in the form of messages.

As part of development, we have integrated the data scraped from the Reddit API in Project-1. Upon parsing the collected dataset, we have indexed it with following schema using Apache Solr:

```
"add-field": [
    {
        "name": "index",
        "type": "string",
        "multiValued": False
    },
    {
        "name": "title",
        "type": "text_en",
        "multiValued": False
    },
    {
        "name": "comments",
        "type": "text_en",
        "multiValued": True
    },
    {
        "name": "topic",
        "type": "string",
        "multiValued": False
    }
]
```

The implemented chatbot will interact with the Solr's dataset and will return appropriate responses to the user's query. The user query can have a normal chit-chat conversation or it can be based on a list of the below 5 topics. The chatbot can distinguish whether the user is talking chitchat or not with an accuracy of 88%.

- Education
- Environment
- Healthcare
- Politics
- Technology

In the case of conversing with the chatbot based on topics, the user can pick one of any given topics or can choose all, considering his choice of desired output.

2. Methodology Chatbot Architecture:

2.1 Models and Data used or Pickled

- Chitchat Dataset
  The Chitchat dataset consists of 7,168 of general conversations
- Reddit Dataset
  The reddit dataset consists of 4000 submissions, each submission has variable number of comments, the total data is around 5,00,000(submissions and comments)

The architecture consists of four models:

- Logistic Regression Model- The logistics regression model is a statistical analysis model to predict a binary outcome. We made use of Logistic Regression in order to classify whether a model belongs to the Chitchat dataset or not.
- Multi-qa-MiniLM-L6-cos-v1- This is a sentence transformer model, which is used to generate embeddings of 384-dimensional space. In our system, this model is labeled as model_embed, which is trained on the chitchat embeddings and the reddit embeddings.
- Principal Component Analysis pickle- The PCA model is used for reducing the dimensionality of embeddings.
- Cross-encoder/Ms-macro-MiniLM-L-6-v2- The cross-encoding model is used to score two sentences pairwise. We have used this model to improve the accuracy. This model is labeled as cross encoder,

The chitchat_embeddings consists of chitchat messages as embeddings, this is loaded as embedding_!, whereas embeddings.pkl consists of reddit comments embeddings which is loaded as comments_data

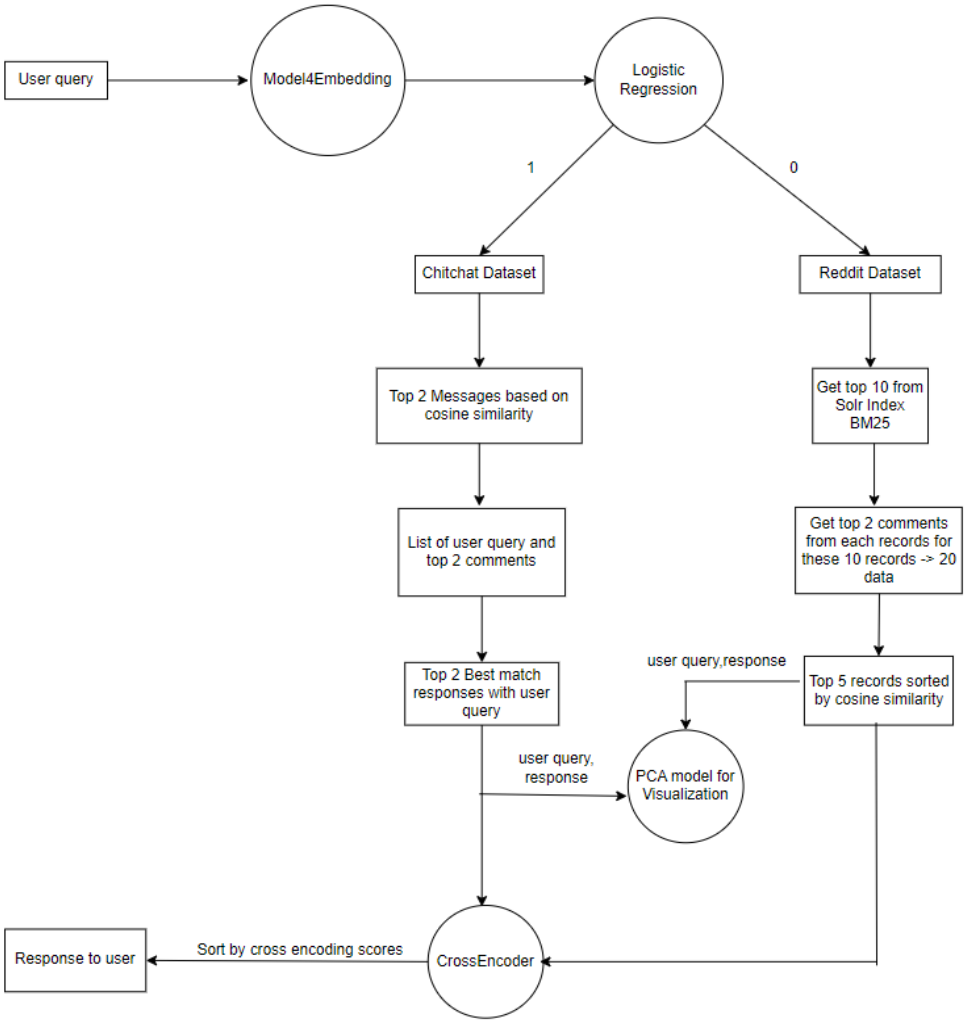
```
array([[ 0.02735302,  0.07566556,  0.09433344, ..., -0.00380192,
        -0.07049689, -0.01487046],
       [ 0.03561354,  0.02801484, -0.03349389, ..., -0.02703307,
        -0.03794186, -0.02942375],
       [-0.00488371, -0.02371128,  0.08396631, ...,  0.00161174,
        -0.04255394, -0.02456819],
       ...,
       [-0.05263273, -0.11659283,  0.01674477, ...,  0.01339425,
         0.12068021,  0.02187884],
       [ 0.03631393, -0.06670718,  0.01304347, ...,  0.0840719 ,
        -0.00572159,  0.000827  ],
       [-0.0408361 ,  0.00217974,  0.0396091 , ..., -0.13142446,
        -0.03508605, -0.0093545 ]], dtype=float32)
```

chitchat_messages.npy

	index	topic	title	comments	embeddings
0	0	politics	'Yes Sir': Sean Hannity Took Direct Orders Fro...	[And all the other stations thought the Clinto...	[[-0.005654253, -0.072430156, -0.011619906, -0...
1	1	politics	Marjorie Taylor Greene told Mark Meadows that ...	[I must've missed that episode of, "Bonanza", ...	[[0.00842428, -0.029089915, 0.043709617, -0.01...
2	2	politics	United States government warns of fears of hou...	[Physical silver has been my game, but when I ...	[[-0.03440206, -0.021355055, 0.034396227, 0.08...
3	3	politics	Trump brazenly asks Putin to release dirt abou...	[Well, maybe they'll manage to legalize pot an...	[[7.558307e-05, 0.04752315, 0.05969657, -0.068...
4	4	politics	Bolton says he recalls Trump using the term "b...	[I'm pretty sure I can guess, without looking ...	[[-0.029306833, 0.0113233905, -0.030916626, -0...
...	...	...	...	...	...
3995	4812	animals	Does Dogs & Cats Help with Depression???	[yes, master language, Pizza time, english bru...	[[-0.010962693, -0.015953913, -0.037021372, 0....

embeddings.pkl

2.2 Implementation



First, the user query is passed to a pre-trained embedding model. We used multi-qa-MiniLM-L6-cos-v1 model for this as it has been trained on a large dataset and finds relevant passages given a query. Also, it was faster to load and performed at similar accuracy as better and larger models. We call this model model_embed for further use purposes to avoid confusing with other models

We store a dataframe that contains all the embeddings of comments as a numpy array, this makes it very easier to access the data once the server has started and dataframe has reloaded, instead of encoding a huge vector again and again.

Similarly, we make an embedding of all the message dataset and store it in a single numpy array. Once we get the top 2 index from it, we can use the message dataset list to retrieve the array. In python, accessing a element in list takes O(1)

So, we merge all these embeddings with y=1 as chit chat and y=0 as not chit chat and train a logistic classifier with it. After training, we get a test accuracy of 88%.

Now using this logistic classifier, we find if it belongs to 1 (chitchat) and 0(not chitchat)

2.2.1 Handling Chit Chat Data

Now we have the following data here:

- We have a list of all messages from the message dataset, then
- We have the embeddings of all comments stored as numpy array weights, which are imported at the start of the server.
- We also have the user query embedding with us.

In the message dataset, we noticed that taken two consecutive messages from the dataset, the second message is mostly the response to the first message. We use this property below:

-> We do a cosine similarity of user query embedding with the entire set of message weights and retrieve the top 2 results and indices of where it's located.

-> We get the sentences of the following indices of both the top results from the message dataset and send it to the cross encoder for post processing.

-> Like for example, How are you matches and returns [28, 500]. We get the next two indices, [29,501] and send it to the cross-encoder for further query.

2.2.2 Handling Reddit Data

In here, we have the following data:

- An embedding dataframe which can be retrieved usefully
- The SOLR url
- The user query embedding.
- A user query and topic

Firstly, we take the user query, topic and return the top 10 documents with highest score in Solr as response. In each of these documents, we have multiple comments which are greater than 100.

We now do the cosine similarity of the query to these multiple comments and return the top 2 comments for every document. After which, we have 20 comments based on the best cosine similarity score and we only return the top 5 of it.

However, to further improve the relevancy and get the best hit, we pass these to the post processor.

2.2.3 Postprocessor

Both Reddit data and Chit chat data are treated in the same way here.

For postprocessors, we use a pre-trained cross encoder model (Ms-macro-MiniLM-L-6-v2).

We pass a user query, hit for all the hits we get as a list of lists and send it to the cross-encoder to compute the cross ranking score. Then, we sort the number hits with a cross encoder score and return the best cross ranking score as the response.

We observed that the cross-encoder ranks better for chitchat than reddit data.

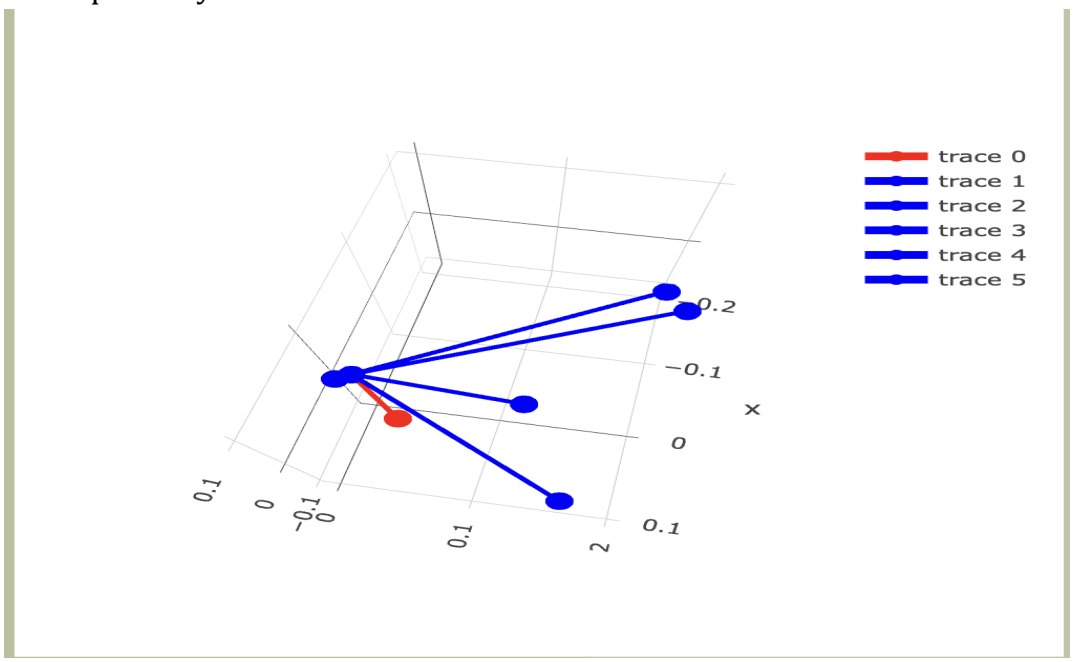
2.2.4 Encoding for Visualization

We pass the user query embedding and the top 5 cosine similarity scores from matches with query to encoder. The encoder uses a non-linear kernel type PCA. We also trained an auto-encoder but we decided to go with PCA as it was faster to compress the data and retrieve it in real time. The PCA reduces the dimensions of vectors from 384 to 3 dimensions. We pass this to Plotly and there we visualize these queries and nearest cosine vectors

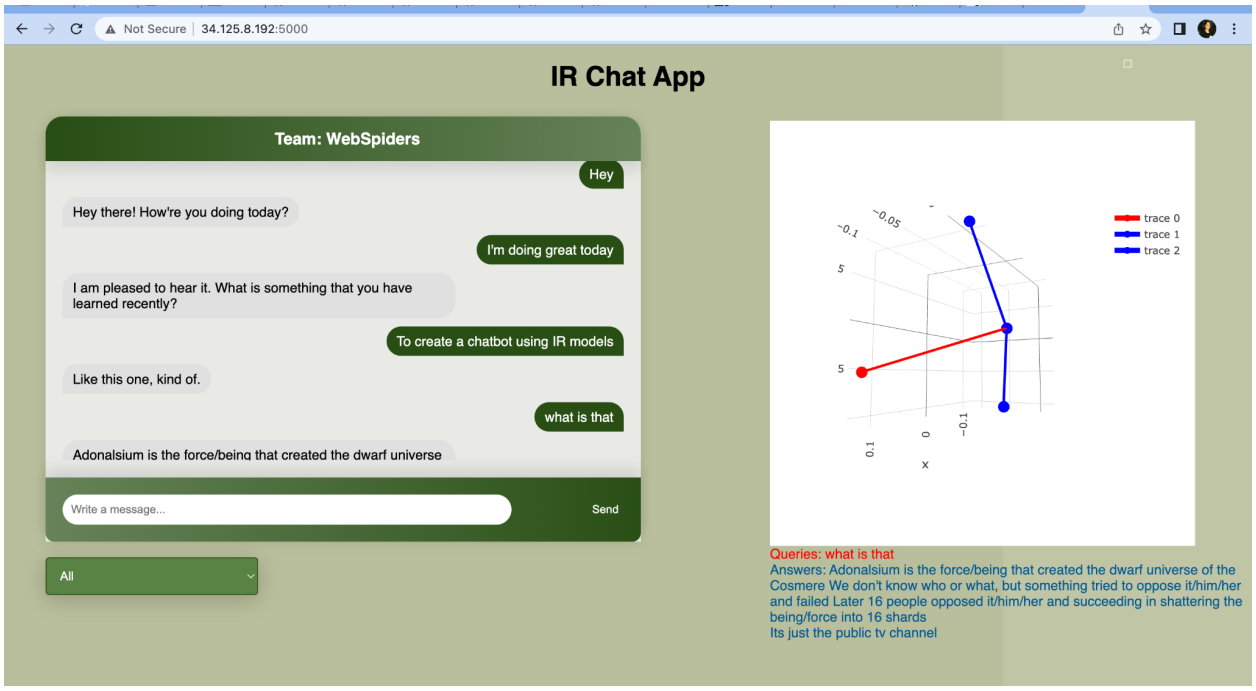
3. Chatbot and Visualization Screenshots:

Web Url: http://34.125.8.192:5000/

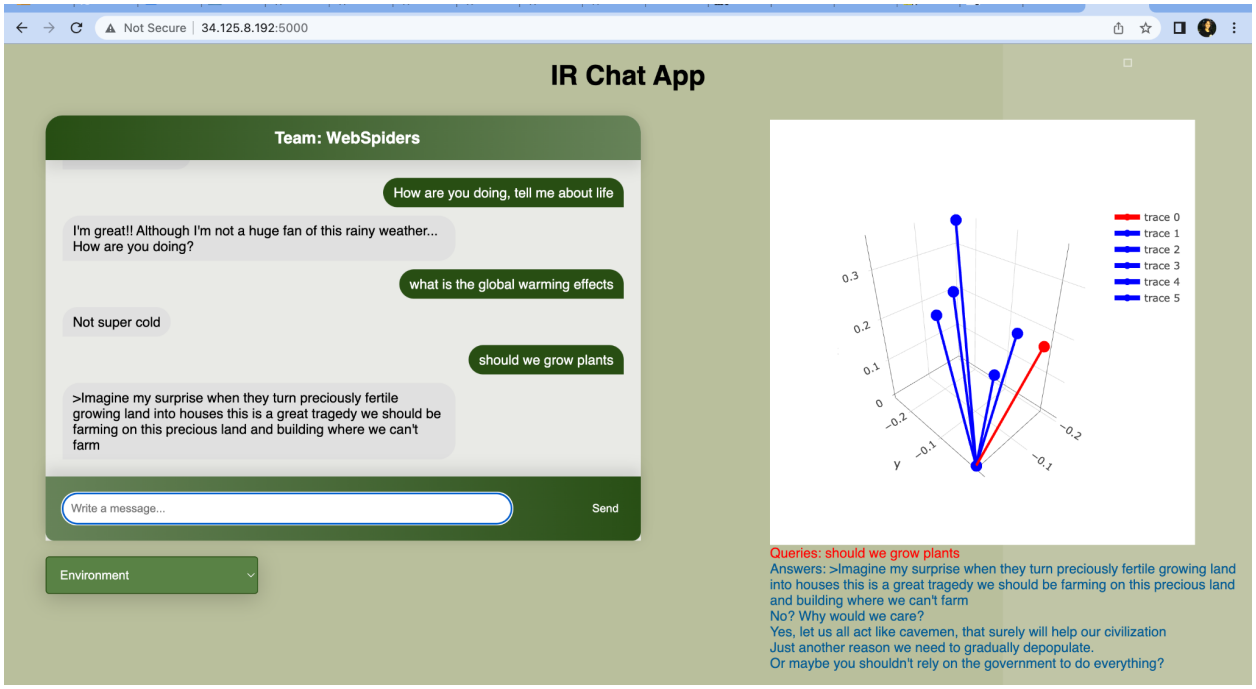
Note: The red and blue arrows in the 3-D graphs denote the user query and chatbot responses respectively.



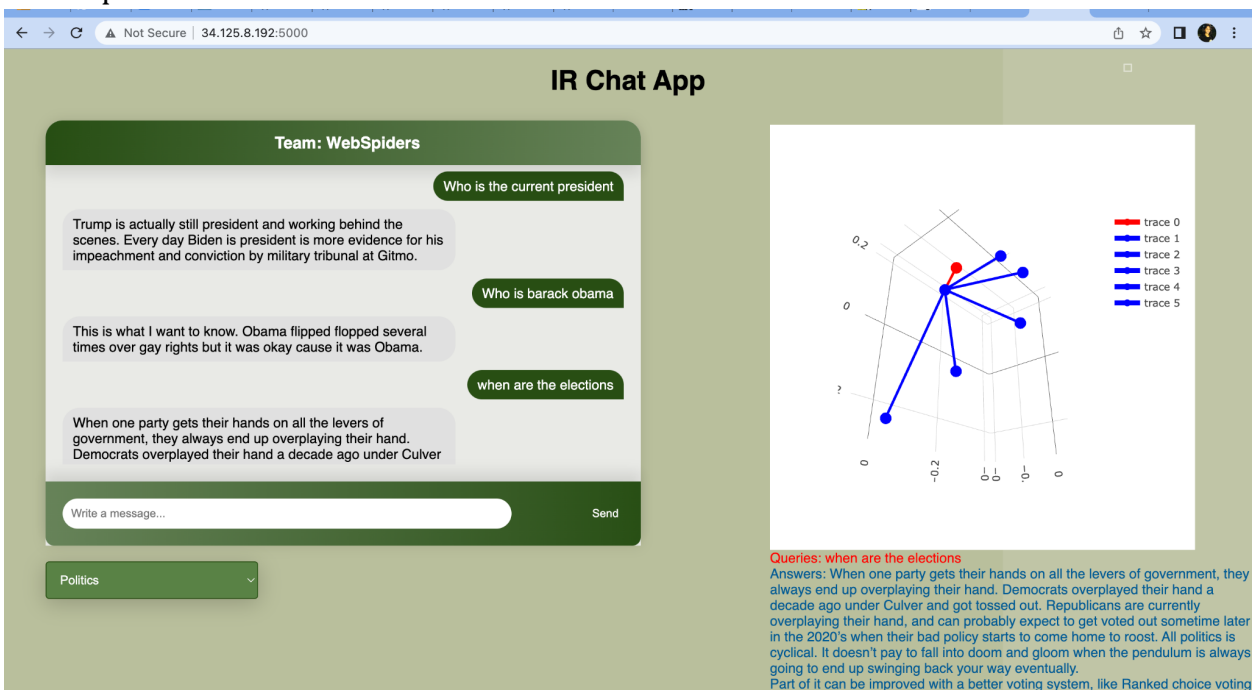
1. With topic: **All**



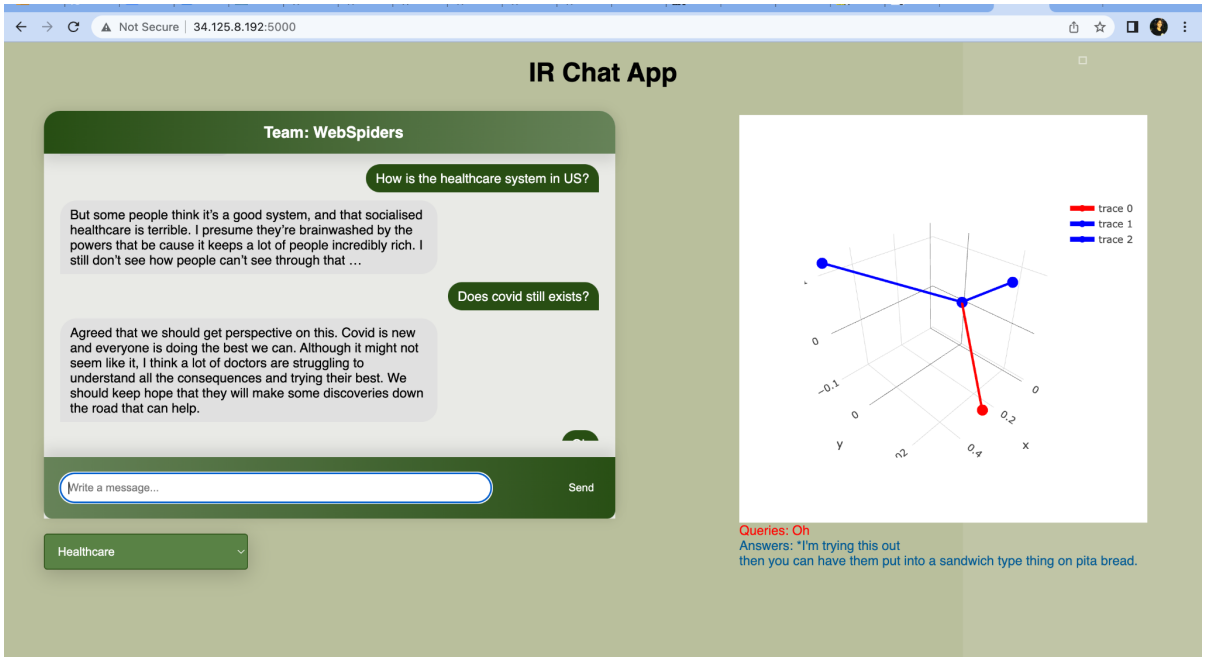
2. With topic: **Environment**



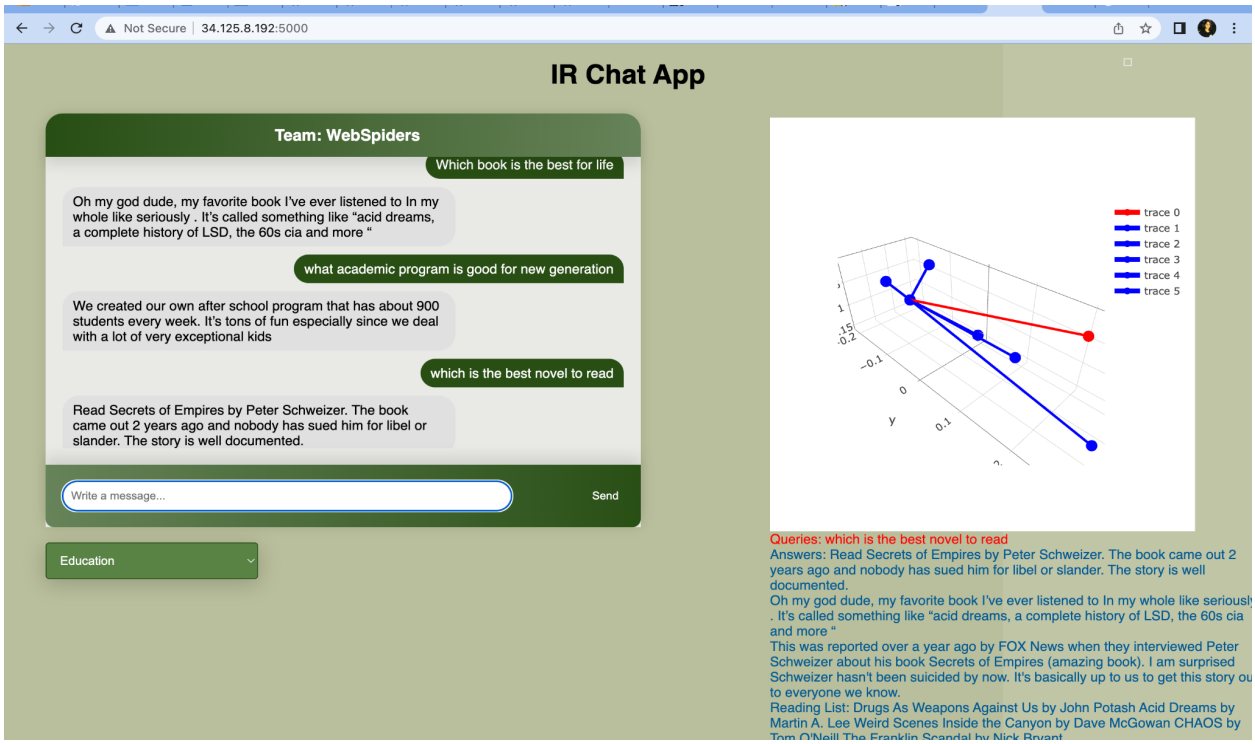
3. With topic: **Politics**



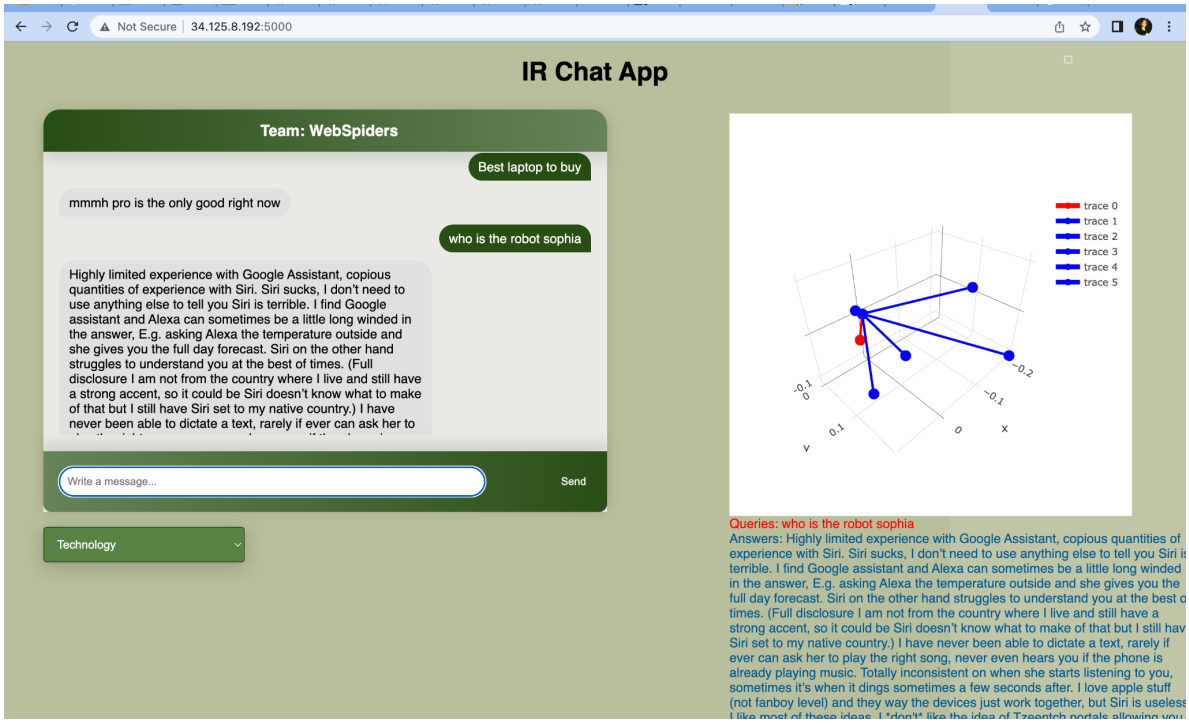
4. With topic: **Healthcare**



5. With topic: Education:



6. With topic: Technology:



4. Work Breakdown by teammates:

1 - Mohit Sai Aravind Nunna → Core Logic to return the best response, handling the backend

2 - Saad Ahmed → Data pre-processing, Logistic classifier for Chitchat, Visualization

3 - Yasmeen Mohammed → UI and Model training

4 - Abhigna → Report, Indexing, Optimization

5. Conclusion:

1. We propose an information retrieval system that can search the reddit dataset for answers. We implemented an IR chatbot that can interact with an end user and return relevant responses for a certain amount of time.
2. Scraping and labeling data plays a significant role in increasing the accuracy of the IR model even more so than training the model