Project - 4

Multi-topic Information Retrieval Chatbot

The Code Linguists

Introduction

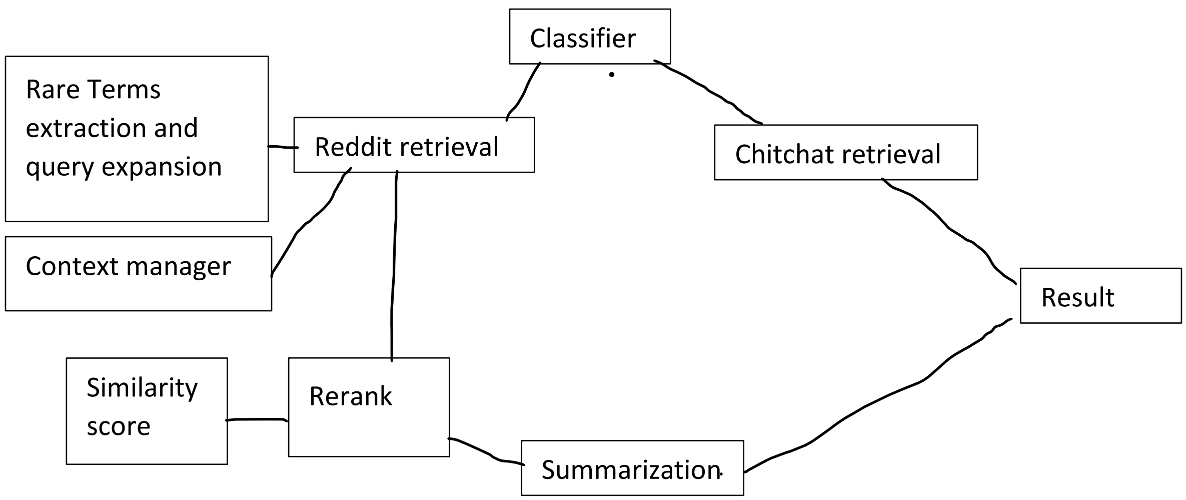
In this report we present our Multi-Topic Information Retrieval chatbot that caters to requests about information available on various subreddits on Reddit. The chatbot is designed to respond to general user queries on the whole knowledge base and focused queries that are directed to a restricted scope in the knowledge base. Restricting the scope of search serves two purposes; First, It can lead to more relevant information retrieved for the user since user's intent is inherently captured when a particular topic is chosen. Second, it serves as an evaluation measure of the chatbot by allowing to capture the global statistics on the efficiency of the model. It creates the foundation for analytics. Healthcare, Environment, Education, Politics and Technology are the topics that are made available for the user to choose from while interacting with the chatbot. Inferring context and tracking is a major challenge and is a necessary component for building a robust chatbot. Our chatbot includes reasonable heuristics to detect context, track and flush context through the chain of conversation. The chatbot is made available as a web application hosted on the cloud.

Chatbot Workflow

1) Chatbot tab -
    a. This subsection provides the user with the graphical user interface to retrieve information from the chatbot.
    b. Users type their message in the Textbox and submit the query using the 'Submit' button; the chatbot comes up with the most relevant response which is shown in the same user interface in a conversational style.
    c. Users can select a persona for the chatbot and the chatbot will align with the selected personality for general chitchat queries. Users can also select a specific topic from the reddit dataset which will act as a filter for retrieval of documents. If no facet is selected, the chatbot searches the entire dataset by default.
    d. Users can reset their session by clicking the 'Clear' button or by refreshing their webpage.
    e. Users can also submit feedback for their last retrieved response by the chatbot, viz. Satisfactory or Not Satisfactory. This helps us to understand the overall performance of the chatbot on different topics in our dataset.
2) Analytics tab - Important statistics and information regarding the last retrieved response are displayed here. Some of the significant statistics that are displayed here for each response are searched index, classifier probability for the searched index, boosted terms (rare terms and context terms) along with displaying the top 50 retrieved documents.

Visualizations tab – This tab displays some interesting visualizations which allow us to infer how well our chatbot is performing on user queries that have their responses originating from the chitchat dataset and the reddit dataset. We can estimate how satisfied the user is with each retrieved document. The visualizations for this subsection represent the global conversation history of the chatbot so far across all users.

Methodology



Data scraping and preparation

The knowledge base is built on the data scraped from Reddit. Data was scraped using the Pushshift API. We've combined all our data to create the corpus. Data is cleaned, i.e filling any missing values by extracting them again and removing useless records. Additionally, we've also used a chitchat dataset**{cite}** to handle general user queries. Our corpus consists of around 100k records from Reddit and around 10k records from Chitchat dataset. The data was indexed into 2 SOLR standalone cores, one consisting of the Reddit data and the other consisting of the Chitchat dataset. While they could be indexed together, enforcing separability diminishes chances of seeing irrelevant results upon searching in their individual indexes as opposed to searching them in a shared index, which would be more prone to junk irrelevant results. Having separated the data, we are faced with the need for classifying incoming queries to search in respective indexes.

Intent Classifier

For classifying a user's Intent we've used a simple logistic regression classifier, tuned for greater accuracy. The model is trained on the whole of Chitchat dataset (10k records) and 10k records of only Reddit submissions, excluding the comments from the Reddit dataset. This ensures that both the Chitchat dataset and the Reddit dataset are fairly represented to the classifier model. Furthermore, the 10k submissions from Reddit should be representative of the data collected from Reddit, since submissions set the course of discourse. The model takes document embeddings (centroid) computed from the pretrained glove word embeddings. Twitter-6B pretrained embeddings with a dimension of 100 were used for the task. Instead of plainly training the model on the document's centroid which assigns equal weight to every term, the document centroid is computed by multiplying the embedding vector with the corresponding IDF weight of the term. This ensures that the classifier picks up on the nature of Reddit dataset consisting of rarer terms compared to the Chitchat dataset. To achieve this, similar words must be identified and grouped together, so we've lemmatized the words using Spacy POS tagging. The IDF dictionary contains the root words. Therefore, training and classification also is performed on the root words. Testing it on a small left-out subset of corpus data yielded an accuracy of around 93%. The model's learned parameters were extracted and are being used for classifying user intent.

Solr schemas

All the relevant fields are defined and indexed appropriately. The core consisting of Reddit data is configured with BM25 global similarity with parameters for b set to 0.9 and k1 to 1.2. The core consisting of Chitchat dataset is configured with DFR similarity with the *basic model* parameter set to G, *after effect* set to L, *normalization* set to H2 and *c* hyperparameter set to 0.7. Due to the nature of Chitchat dataset, short records and a comparatively small dataset, DFR's capability in handling short queries and small corpuses made more sense. However, no discernable difference was noticed when BM25 was configured the Chitchat core.

```
<schema name="default-config" version="1.6">
  <uniqueKey>id</uniqueKey>
  <similarity class="solr.BM25SimilarityFactory">
    <str name="k1">1.2</str>
    <str name="b">0.9</str>
  </similarity>
```

```
<schema name="default-config" version="1.6">
  <uniqueKey>id</uniqueKey>
  <similarity class="solr.DFRSimilarityFactory">
    <str name="basicModel">G</str>
    <str name="afterEffect">L</str>
    <str name="normalization">H2</str>
    <float name="c">7</float>
  </similarity>
```

```
<fieldType name="text_en_red" class="solr.TextField" positionIncrementGap="100">
  <analyzer type="index">
    <charFilter name="mapping" mapping="cmap.txt"/>
    <charFilter name="patternreplace" pattern="\\" replacement=""/>
    <charFilter name="patternreplace" pattern="US" replacement="USA"/>
    <charFilter name="patternreplace" pattern="([A-Z]).(?=[A-Z])" replacement="$1"/>
    <tokenizer name="standard"/>
    <filter name="phonetic" encoder="soundex"/>
    <filter name="shingle" minShingleSize="2" maxShingleSize="4"/>
    <filter name="stop" ignoreCase="true" words="lang/stopwords_en.txt"/>
    <filter name="lowercase"/>
    <filter name="englishPossessive"/>
    <filter name="keywordrepeat"/>
    <filter name="porterStem"/>
    <filter name="removeduplicates"/>
  </analyzer>
  <analyzer type="query">
    <charFilter name="mapping" mapping="cmap.txt"/>
    <charFilter name="patternreplace" pattern="([A-Z]).(?=[A-Z])" replacement="$1"/>
    <tokenizer name="standard"/>
    <filter name="phonetic" encoder="soundex"/>
    <filter name="stop" ignoreCase="true" words="lang/stopwords_en.txt"/>
    <filter name="lowercase"/>
    <filter name="englishPossessive"/>
    <filter name="keywordrepeat"/>
    <filter name="porterStem"/>
    <filter name="removeduplicates"/>
  </analyzer>
</fieldType>
```

Context Identification and tracking

For managing context, heuristics were used to achieve the desired functionality. We are making a Markov assumption for tracking context, i.e, named entities, if any present, from each query are extracted using Spacy NER feature and the subsequent query is expanded with the entities from the previous query. But context should not be carried forward for each subsequent query. Context can be switched by the user. This is tracked using similarity search between contiguous queries from sentence transformers library, which effectively computes the cosine similarity between sentences based on their embeddings. If the similarity between a contiguous pair of queries meets a defined threshold, context continuity is assumed and the next query is expanded with the previous query's named terms, else the previous query's named entities are flushed. Additionally, when the user switches topics, context is flushed and if the user does not switch topics, context is carried forward. Finally, context is flushed at the end of each session.

Query Formulation

Queries are formulated following the edismax syntax, as it offers a wider set of possibilities when it comes to boosting terms in the queries. Query is formulated based on whether the user chooses a corpus-wide general query or a topic-wise faceted search. Faceted search functionality is achieved using the filter query parameter. When formulating a query, both the question-and-answer fields are searched. Retrieval is based on searching the question, where rare terms and context terms are boosted and reranking is performed by searching the answer field for rare terms and context terms. Context terms refer to the named entities extracted from the previous query. Rare terms are the 2 rarest terms in any given query. These rare terms are further expanded by finding 3 top similar words to each of them from pretrained Glove embeddings.

{show a query example}

Scoring

For retrieval, the documents are scored following the configured similarity rules, which is output by Solr. For reranking, the scores returned by solr and another additional similarity score computed between the query and response based on the pretrained embeddings is used.

Score = (1-alpha) * similarity score + alpha * BM25 score.

Alpha was set to 0.7 after tuning it based on subjective perception of the produced results.

No reranking is performed on results retrieved from Chitchat datasets. It performs well with basic retrieval given the nature of the data and user queries.

Summarization

Finally, the text from the best response is summarized so that useful information is captured and presented to the user. Pretrained BERT models for text summarization are used to achieve this.

Analytics

The analytics tab in the UI is designed for explainability. It also helps with debugging. It contains information on the index being searched, classifier output, rare terms being searched, context terms being searched. BM25 score, similarity score, total score, complete text of the summarized response presented to the user, and all the documents retrieved for the query.

User Feedback

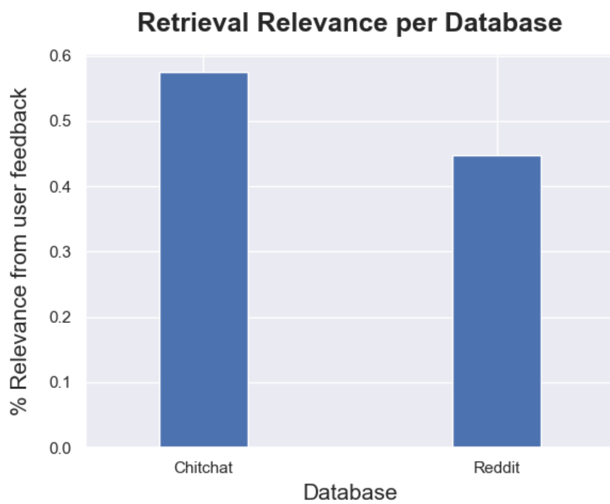
The app also has provisions for user feedback. The user can provide feedback on whether the results were relevant to their information need. Currently we are using user feedback for the purpose of visualization.

Database

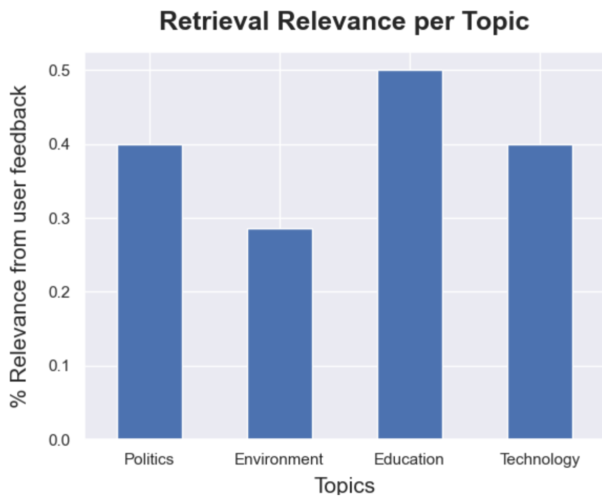
The app stores data on each turn in a MysSql server. Session id, User id, query, best response, all responses, scores, faceted topics, bot personality and user feedback are stored in the database and will be used for creating visualizations that explain the performance of our model.

Visualizations

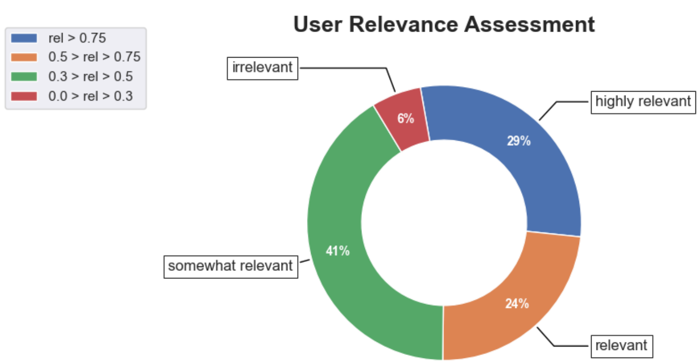
By allowing the user to provide feedback for each response, we generate several statistics that indicate the performance of our chatbot across different topics, databases and from the perspectives of different users.



Aggregating the feedback received for each database queried, we calculate the relevance w.r.t each. We see that, when querying the chitchat dataset, more relevant responses have been found as compared to 'Reddit' dataset. This implies that Reddit dataset is more complex and must cater to more diverse information needs making relevant retrieval difficult.



Another statistic calculated is the relevance for each topic when user has turned on faceted search. The above chart shows that chatbot performed the best when topic was ‘Education’, followed by ‘Technology’ and ‘Politics’. ‘Environment’ related queries were the hardest for the chatbot.

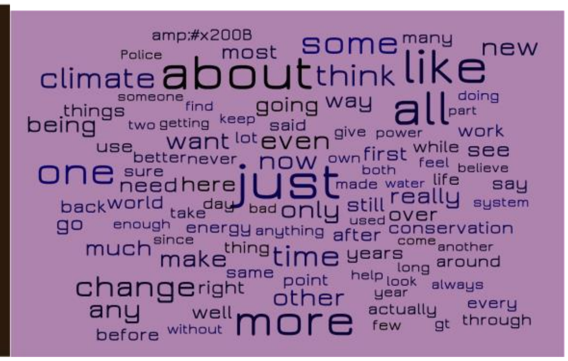


Lastly, to measure the overall satisfaction of users and their information needs, we aggregate data for 20+ different chat sessions. We notice that majority of the users found the responses somewhat relevant. Almost 1/3 of the users found the chatbot’s responses highly relevant (more than 2/3 of the responses in a conversation marked relevant). Further only 6% of total users did not have their information needs met and found most of the responses in their conversation irrelevant.

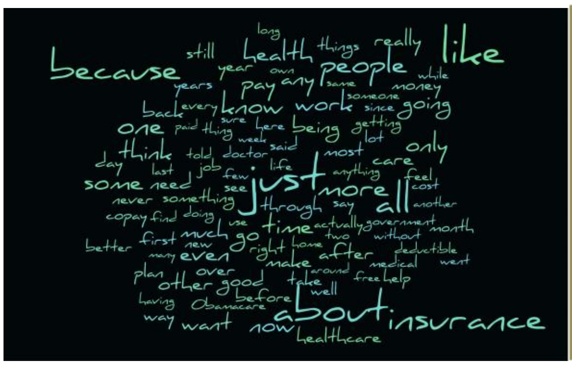
Lastly, to understand our dataset and the most retrieved documents across different topics, we generate word clouds.

Education

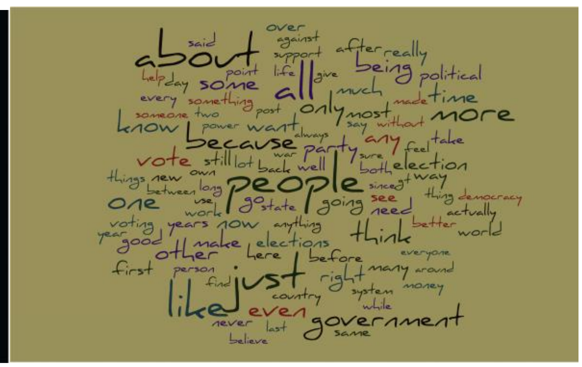
Environment



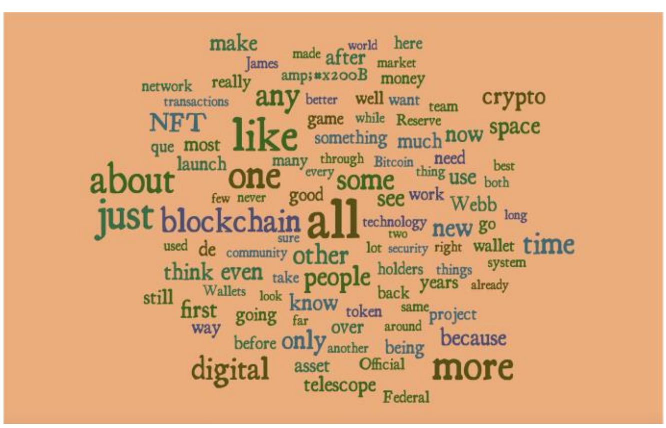
Healthcare



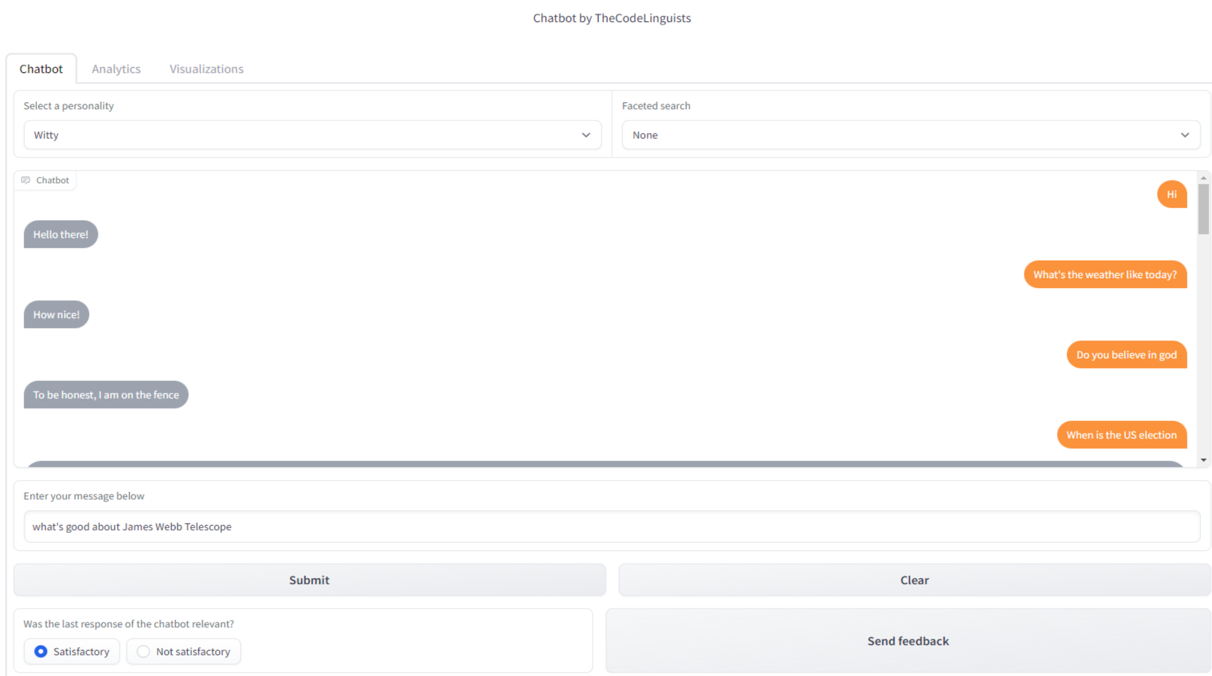
Politics

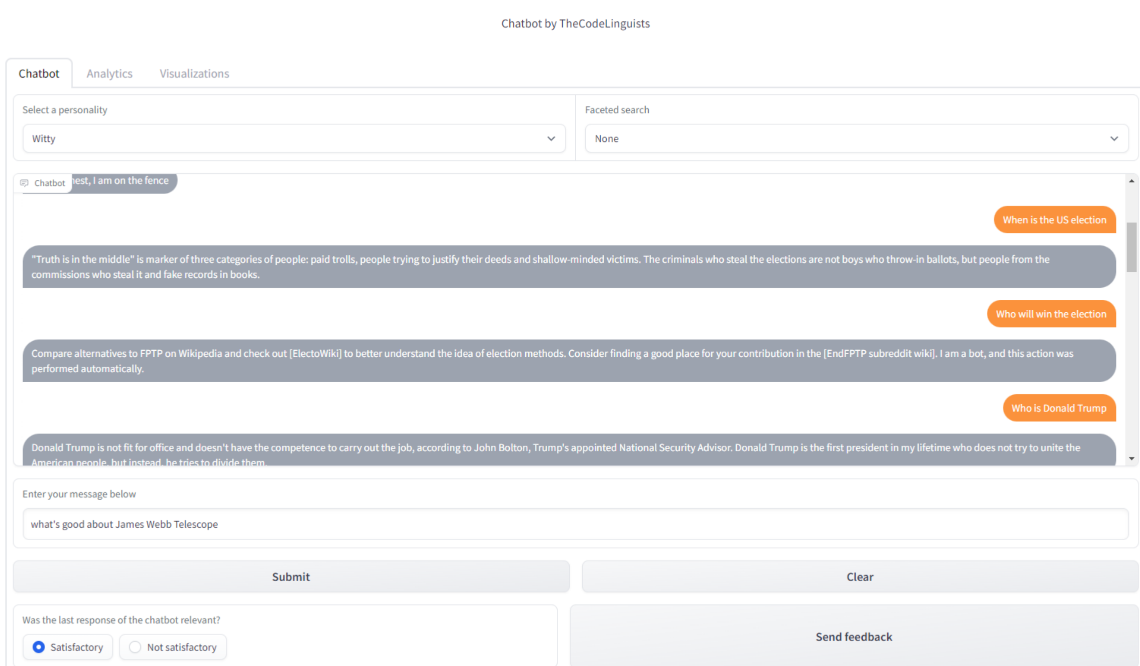


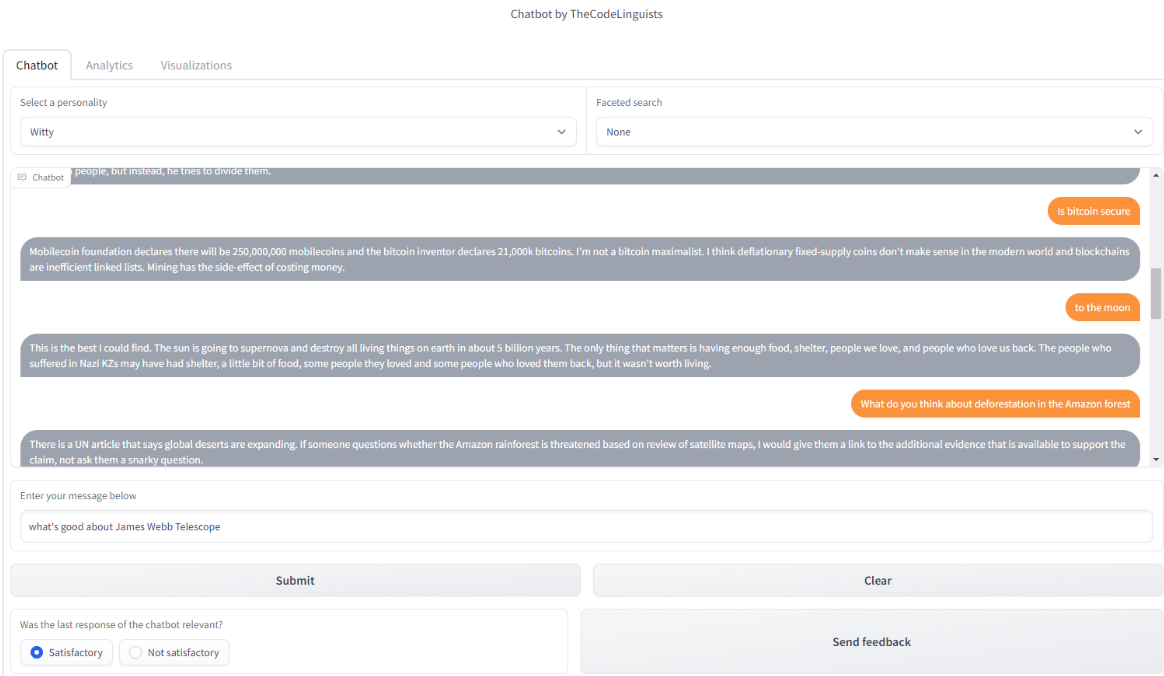
Technology

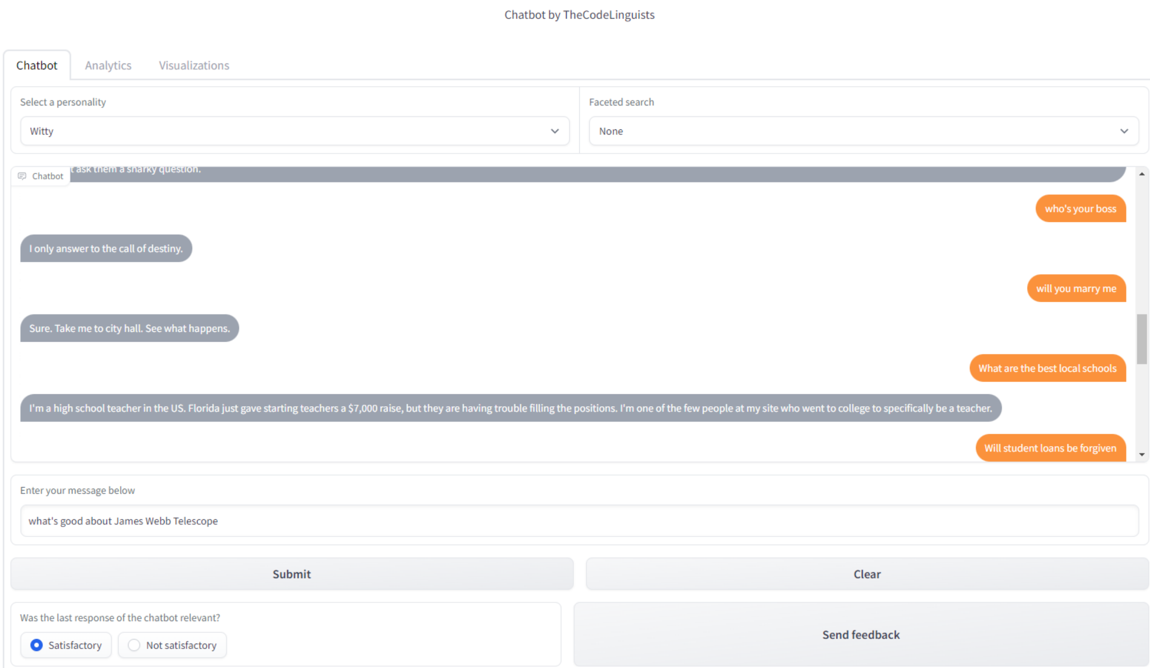


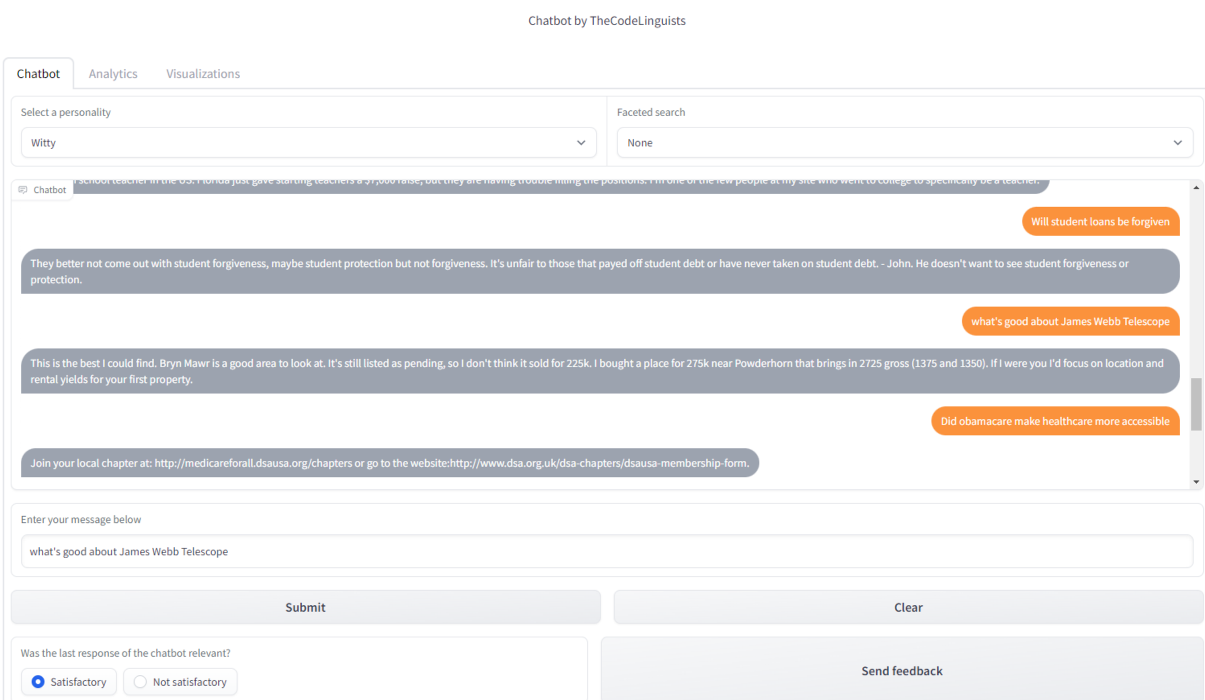
Example (Chat Screenshots)

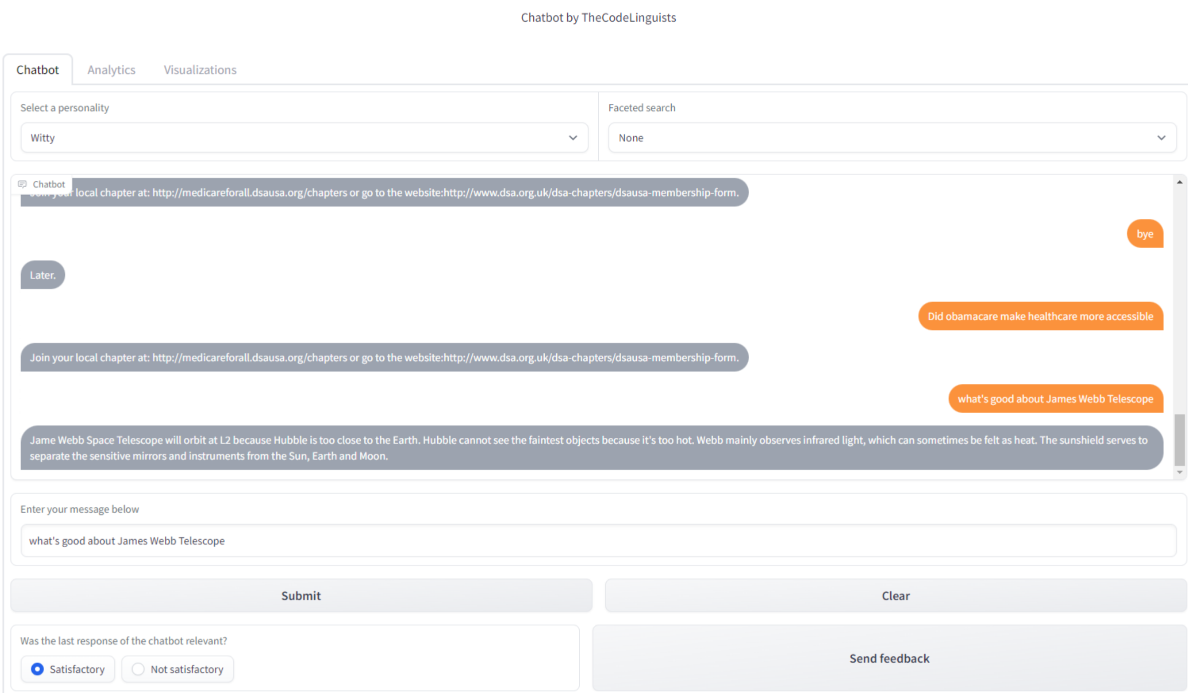





Chatbot by TheCodeLinguists
Chatbot
Analytics
Visualizations
Select a personality
Witty
Faceted search
None
Chatbot
people, but instead, he tries to divide them.
Is bitcoin secure
Mobilecoin foundation declares there will be 250,000,000 mobilecoins and the bitcoin inventor declares 21,000k bitcoins. I'm not a bitcoin maximalist. I think deflationary fixed-supply coins don't make sense in the modern world and blockchains are inefficient linked lists. Mining has the side-effect of costing money.
to the moon
This is the best I could find. The sun is going to supernova and destroy all living things on earth in about 5 billion years. The only thing that matters is having enough food, shelter, people we love, and people who love us back. The people who suffered in Nazi KZs may have had shelter, a little bit of food, some people they loved and some people who loved them back, but it wasn't worth living.
What do you think about deforestation in the Amazon forest
There is a UN article that says global deserts are expanding. If someone questions whether the Amazon rainforest is threatened based on review of satellite maps, I would give them a link to the additional evidence that is available to support the claim, not ask them a snarky question.
Enter your message below
what's good about James Webb Telescope
Submit
Clear
Was the last response of the chatbot relevant?
Satisfactory
Not satisfactory
Send feedback


Chatbot by TheCodeLinguists
Chatbot
Analytics
Visualizations
Select a personality
Witty
Faceted search
None
Chatbot
who's your boss
I only answer to the call of destiny.
will you marry me
Sure. Take me to city hall. See what happens.
What are the best local schools
I'm a high school teacher in the US. Florida just gave starting teachers a $7,000 raise, but they are having trouble filling the positions. I'm one of the few people at my site who went to college to specifically be a teacher.
Will student loans be forgiven
Enter your message below
what's good about James Webb Telescope
Submit
Clear
Was the last response of the chatbot relevant?
Satisfactory
Not satisfactory
Send feedback


Chatbot by TheCodeLinguists
Chatbot
Analytics
Visualizations
Select a personality
Witty
Faceted search
None
Chatbot
Will student loans be forgiven
They better not come out with student forgiveness, maybe student protection but not forgiveness. It's unfair to those that payed off student debt or have never taken on student debt. - John. He doesn't want to see student forgiveness or protection.
what's good about James Webb Telescope
This is the best I could find. Bryn Mawr is a good area to look at. It's still listed as pending, so I don't think it sold for 225k. I bought a place for 275k near Powderhorn that brings in 2725 gross (1375 and 1350). If I were you I'd focus on location and rental yields for your first property.
Did obamacare make healthcare more accessible
Join your local chapter at: http://medicareforall.dsausa.org/chapters or go to the website:http://www.dsa.org.uk/dsa-chapters/dsausa-membership-form.
Enter your message below
what's good about James Webb Telescope
Submit
Clear
Was the last response of the chatbot relevant?
Satisfactory
Not satisfactory
Send feedback



Observations

Above is a string of 17 query responses from the chatbot with a few of them invoking faceted search. The chatbot does fairly well when the queries correspond to the domain of the data scraped. Reranking is not flawless but works in most cases. There is a potential risk of query drift when expanding the query based on similar words to the rare words in query. This could lead to irrelevant results. Facet search also works fairly well; sometimes it improves the relevancy of the responses, at other times, it leads to fetching irrelevant information. A trivial workaround would be to simply fetch documents boosted on the rare terms from the query. While this could work, it does not generalize well when catering to the user's information need. It may or may not improve precision depending on the query, but it will diminish the recall of the system. Lastly, the chatbot has a latency of between 5-10 seconds when fetching results from the Reddit index. This is due to the complexity of the model which has been described above. We wanted to build a complex model which would present a greater opportunity to learn.

Alternatives explored

We tried to implement Dual Embedding Space Model, but we've hit a snag with accessing output layer embeddings on pretrained Glove models. You can use the out-layer embeddings by training on the data available, but the performance wasn't good enough compared to the pre-trained embeddings.

We explored Solr's Learning to Rank feature but realized we don't have the relevance judgement data to achieve that. We contemplated constructing random queries from the corpus and using the similarity scores as a proxy for relevancy, but it seemed like an inefficient approach which would take a long time to build and train the models with random queries.

For context management, we've tried basing the context continuity based on the similarity between the query and previous response, but this has proven to be ineffective compared to taking the similarity between contiguous query pairs. Due to the size of the responses, and presence of many other terms, similarity is not being robustly captured.

Contribution

Praneeth Nekkalapudi

Data scraping/cleaning, Solr schema configuration and indexing, Implemented the classifier, retrieval, reranking and context tracking logic.

Eva Pradhan

Integrated the Database.

Sumedh Khodke

Frontend and Backend for the chatbot.

Jay Lal

Visualizations, Deployment, Assisted in designing classifer and context logic.

Conclusion

We were able to implement a decent chatbot, that performs several actions which we've learned in the course (Indexing, Retrieving, Ranking, Reranking, Summarization, Embeddings, Lemmatization, POS tagging, Named Entity detection, boosting, indexing and query filters). The chatbot caters to the information need of the user moderately well. Plenty of scope to improve the model in every area, especially context tracking and reranking which will be explored in the future.