Shaan Sekhon Report Chatbot (2)

Data Collection and Processing

Web Scraping: The system starts by scraping a starter URL using `requests` and `BeautifulSoup` to parse HTML content. It extracts links that contain keywords related to Neymar and checks these links for accessibility before saving them to a file. This initial collection is designed to gather a diverse set of sources for comprehensive data analysis.

Data Cleaning: Extracted text is cleaned to remove HTML tags, special characters, and irrelevant segments using custom cleaning functions and regular expressions. This step is crucial for preparing the text for tokenization and further analysis.
Text Analysis and NLP Techniques

Tokenization and Lexical Analysis:** Utilizes NLTK's tokenization methods to break text into sentences and words, which are then processed to remove stopwords and non-alphabetical characters. This cleaned list of words is used to compute lexical diversity, indicating the variety of vocabulary used in the text.

TF-IDF Analysis: The system computes the Term Frequency-Inverse Document Frequency (TF-IDF), a statistical measure used to evaluate how important a word is to a document in a collection or corpus. This analysis helps in identifying key terms and their significance across multiple documents.

Sentiment Analysis: Incorporates NLTK's SentimentIntensityAnalyzer to assess the sentiment of the text. This feature is particularly useful for understanding public sentiment or opinions expressed in the articles about Neymar.

System Description for the Chatbot Training Model

The system is designed for training a chatbot using a recurrent neural network (RNN) with long short-term memory (LSTM) units. The model is developed using Python libraries such as NumPy, PyTorch, NLTK, and others to process text data, learn from it, and effectively perform text classification. This classification capability enables the chatbot to understand user intents and respond appropriately.

Data Processing and Enhancement

Text Processing: The system employs text processing techniques using the `nltk` library, which includes tokenization (breaking down sentences into words or tokens) and lemmatization (reducing words to their base or dictionary form). This preprocessing aids in normalizing the data before training.

Data Augmentation with Synonyms: To enhance the training data, the system expands each sentence in the training data with synonyms, using the WordNet database via NLTK. This approach aims to make the model robust to variations in user input, enhancing its ability to generalize across similar phrases with different wording.

Machine Learning and NLP Techniques

RNN with LSTM: The core of the model is an LSTM-based RNN. LSTMs are well-suited for sequence prediction problems because they are capable of storing past information, which is crucial for understanding context in language.

Word Embeddings: Inputs to the LSTM are transformed into embeddings, which are dense vector representations of words that capture their semantic properties. This system employs a simple form of embeddings generated by a bag-of-words model, as indicated by the use of a custom `bag_of_words` function. Bag-of-words models count word occurrences but typically do not capture the order of words.

Training and Optimization: The model is trained using the cross-entropy loss function and optimized with the Adam optimizer. Additionally, a learning rate scheduler reduces the learning rate periodically to fine-tune the training process.

System Architecture

Data Loader: Data is encapsulated in a custom `Dataset` class, which is then loaded using PyTorch's `DataLoader`. This setup provides efficient, batch-wise iteration over the dataset during training.

Training Loop: The model is trained over multiple epochs, where in each epoch, the model processes batches of data, computes loss, and updates model parameters.

Evaluation: The system includes an evaluation function to compute the confusion matrix using Seaborn and Matplotlib for visualization. This matrix helps in understanding the model's performance across different classes.

Usage of External Libraries
- NumPy for handling numerical operations on arrays.
- PyTorch for building and training the neural network.
- NLTK for natural language processing tasks.
- Matplotlib and Seaborn for visualizing the performance of the model.
- SciKit-Learn for creating confusion matrices and splitting the dataset.

This system is comprehensive in its approach to building a robust chatbot model by utilizing modern techniques in machine learning and natural language processing, making it capable of understanding and responding to user inputs with a significant degree of intelligence and relevance.

Additional Features and Implementation Details

User Interaction:
CSV File Operations: The system includes functions to read from and write to CSV files. This capability is used to store and retrieve user data, such as names and interaction dates. This allows the system to maintain a record of user interactions, which can be useful for personalized user experiences.

User Interface and Experience:
Interactive Chat Interface: The system prompts the user for input and provides responses based on the model's predictions. It introduces a chatbot named "Santos", designed to discuss topics related to the football player Neymar.
Real-Time Data Fetching: The system integrates real-time data fetching from external sources (like ESPN for soccer standings) to provide up-to-date information during the interaction, enhancing the user experience with live data.
Input Validation and User Feedback: The system seeks user feedback on the responses given, which can be used to gauge user satisfaction and improve the model's performance indirectly by adapting based on user interaction patterns.

Web Scraping:
Integration with Google Search: The system uses the `googlesearch` Python library to perform web searches based on user queries, which allows the bot to provide additional resources when it cannot directly answer a question. This feature helps in maintaining an engaging user interaction by reducing the instances where the bot fails to provide a satisfactory answer.

Model Deployment and Operational Aspects:
Device Compatibility: The model is set up to run on either GPU or CPU, depending on availability, which ensures that the system can be deployed flexibly across various hardware setups.
Model State Management: The system includes mechanisms to load and save the model's state, ensuring that the training progress and model parameters can be preserved across sessions.

Enhanced Machine Learning and NLP Techniques

Advanced User Data Management: By incorporating user interaction history into the model's operational logic, the system can tailor its responses based on past interactions, potentially increasing user engagement and satisfaction.

Dialogue Logic

```
Santos: "Hi! I am Santos, a Neymar fan bot! I love to talk about Neymar."
        "If you want to quit, just type quit and press enter!"
        "What is your name?"
User: (provides name or types "quit")
    ├─── "quit" ──> System: "BYE!" (exit)
    └─── (provides name)
        ├─── System checks if user is returning or new
        │    ├─── Returning User ──> "Welcome back, [name]!"
        │    └─── New User ──> "Hi, [name]!"
        └─── "Would you like me to pull a Live Table and check to see how Neymar's team is doing now?"
            User: (response or types "quit")
                ├─── "quit" ──> System: "BYE!" (exit)
                └─── (response)
                    ├─── System checks sentiment of response
                    │    ├─── Positive ──> Shows Live Table
                    │    │    ├─── "If you want to see the stats press 1, else press 2"
                    │    │    │  User: (chooses 1 or 2 or types "quit")
                    │    │    │       ├─── "quit" ──> System: "BYE!" (exit)
                    │    │    │       ├─── 1 ──> Shows additional stats
                    │    │    │       └─── 2 ──> Ends current function
                    │    │    └─── System updates user interaction about Neymar's team's ranking
                    │    └─── Negative/Neutral ──> "No problem!"
                    └─── "Let's chat! (type 'quit' to exit)"
                      "You can ask me anything about Neymar!"
                      User: (asks a question or types "quit")
                          ├─── "quit" ──> Saves user data, System: "BYE!" (exit)
```

```
└──── (asks a question)
   ├──── System processes question, determines response confidence
   |     ├──── High confidence ───> Provides an appropriate response
   |     └──── Low confidence ───> Offers web search results to help
   └──── Occasionally asks user to rate the response satisfaction
     User: (rates or skips)
         └──── Saves feedback
```

Strengths:

Personalization and User Recognition: The bot attempts to recognize returning users by searching for their name in a CSV file. This can help create a personalized experience by recalling past interactions or user preferences.

Sentiment Analysis: Utilizing NLTK's Vader Sentiment Intensity Analyzer allows the bot to gauge the sentiment of user inputs, which could guide the conversation flow or actions (like deciding whether to show certain content).

Dynamic Content Retrieval: The bot can fetch live data (like soccer team standings) from the web, keeping the content it delivers relevant and up-to-date.

User Engagement: By occasionally asking for feedback on the answers provided, the bot can potentially improve its interactions over time based on user satisfaction scores.

Neural Network Integration: Employing a neural network (RNN with LSTM) allows the bot to potentially understand and process a wide range of user queries through trained intents and responses.

Weaknesses:

Dependency on Sentiment for Functionality: The decision to show certain content based only on sentiment analysis might limit the functionality unnecessarily if the sentiment analysis does not accurately reflect user intent.

Limited Understanding of Context: While LSTM helps with sequence understanding, the bot's capability to understand context or manage state across multiple turns in a conversation appears limited. This might lead to less coherent interactions in more complex dialogues.
Reliance on Hardcoded Elements: The use of hardcoded URLs and CSV reading/writing operations could make the bot less flexible and harder to maintain or scale.

Survey:

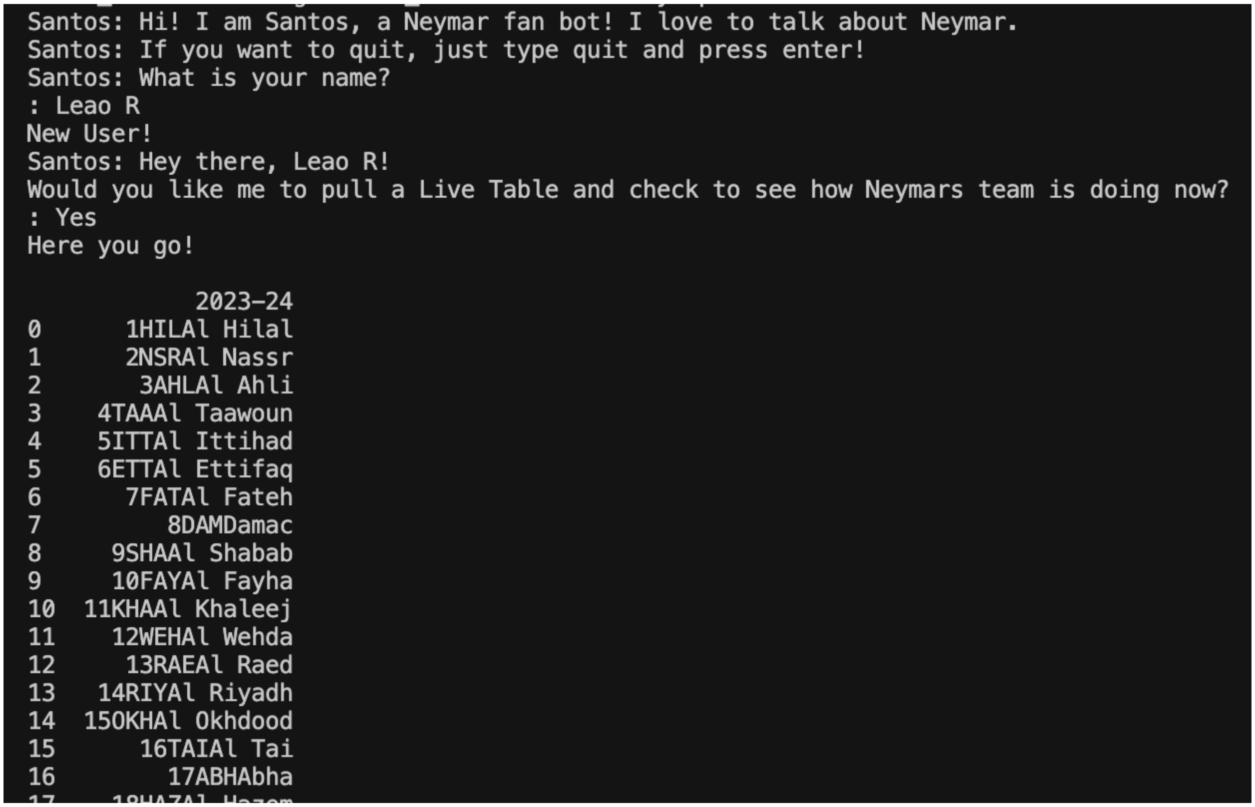
1. How satisfied are you with the interaction you had with our chatbot today?

2. To what extent do you feel the responses from our chatbot were relevant to your queries?

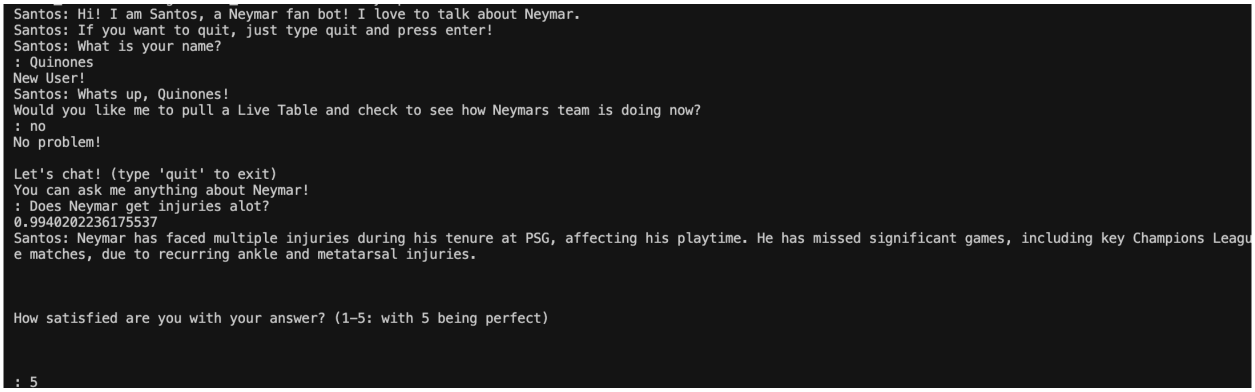
3 .How easy was it to communicate with our chatbot?

1 (Very difficult) 2 (Somewhat difficult) 3 (Neutral) 4 (Somewhat easy) 5 (Very easy)

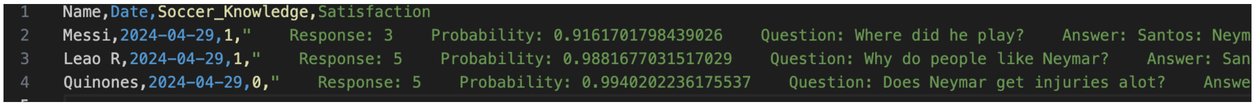
USER 1: 5, 5, 5
USER 2: 5, 3, 5,
USER 3: 4, 4, 4



```
Santos: Hi! I am Santos, a Neymar fan bot! I love to talk about Neymar.
Santos: If you want to quit, just type quit and press enter!
Santos: What is your name?
: Leao R
New User!
Santos: Hey there, Leao R!
Would you like me to pull a Live Table and check to see how Neymars team is doing now?
: Yes
Here you go!

           2023-24
0      1HILAl Hilal
1      2NSRAl Nassr
2       3AHLAl Ahli
3    4TAAAl Taawoun
4    5ITTAl Ittihad
5    6ETTAl Ettifaq
6      7FATAl Fateh
7         8DAMDamac
8     9SHAAl Shabab
9     10FAYAl Fayha
10  11KHAAl Khaleej
11    12WEHAl Wehda
12     13RAEAl Raed
13   14RIYAl Riyadh
14  15OKHAl Okhdood
15      16TAIAl Tai
16        17ABHAbha
17    18HAZAl Hazem
```



```
Santos: Hi! I am Santos, a Neymar fan bot! I love to talk about Neymar.
Santos: If you want to quit, just type quit and press enter!
Santos: What is your name?
: Quinones
New User!
Santos: Whats up, Quinones!
Would you like me to pull a Live Table and check to see how Neymars team is doing now?
: no
No problem!

Let's chat! (type 'quit' to exit)
You can ask me anything about Neymar!
: Does Neymar get injuries alot?
0.9940202236175537
Santos: Neymar has faced multiple injuries during his tenure at PSG, affecting his playtime. He has missed significant games, including key Champions Leagu
e matches, due to recurring ankle and metatarsal injuries.



How satisfied are you with your answer? (1–5: with 5 being perfect)



: 5
```



```
Name,Date,Soccer_Knowledge,Satisfaction
Messi,2024-04-29,1,"    Response: 3    Probability: 0.9161701798439026    Question: Where did he play?    Answer: Santos: Neym
Leao R,2024-04-29,1,"    Response: 5    Probability: 0.9881677031517029    Question: Why do people like Neymar?    Answer: San
Quinones,2024-04-29,0,"    Response: 5    Probability: 0.9940202236175537    Question: Does Neymar get injuries alot?    Answe
```