CS 584 Project Final Report

Multilingual Chatbot with NLP, Deep Learning, and Web Scraping

Shrey Shah (20009523), Hitesh Kardam (20011900), Aman Sandal (20011102)

1. Introduction

1.1 Overview

This report discusses the development of a versatile, multilingual chatbot that integrates Deep Learning, Natural Language Processing (NLP), and web scraping technologies. This chatbot will provide users with a wide range of services, including joke-telling, weather updates, horoscope predictions, answering random queries, and a Sudoku game. This proposal concisely overviews the project's objectives, methods, and expected outcomes. We also discuss existing challenges and related work in the field of conversational AI.

This report presents the progress on the development of a multilingual chatbot designed to offer a range of services including horoscopes, Sudoku, translations, and weather updates. The chatbot utilizes Deep Learning and NLP to interact in 100 languages, aiming to provide an engaging and useful experience for users. Progress has been made in several areas, with notable achievements in horoscope provision in different languages and the implementation of a functioning Sudoku game. Challenges have been encountered in weather data retrieval and sudoku game answering, which are being actively addressed.

1.2 Background

The progression of chatbots, evolving from early rule-based systems to advanced AI-driven models, showcases the dynamic interplay between Natural Language Processing (NLP) and Machine Learning (ML) techniques. These conversational agents have found applications across various domains like customer service, healthcare, education, and finance. The categorization of chatbots into different types, including task-oriented, conversational, rule-based, retrieval-based, and generative, provides insights into their functionalities and complexities.

However, challenges persist, encompassing issues such as ambiguous user queries, context management, and sustaining user engagement. Effective chatbot design is shaped by Human-Computer Interaction (HCI) and User Experience (UX), highlighting the significance of user-centric approaches. Ethical considerations, spanning from bias to privacy and security, require thoughtful consideration in the development process. This literature background underscores the multidimensional nature of chatbot research, covering technological advancements, user-centered design principles, ethical dimensions, and future trends, establishing a robust foundation for the subsequent exploration and implementation within the current project.

1.3 Challenges

The main challenges in our project were ensuring the accuracy of the chatbot's responses, enabling multilingual support, and integrating various services like horoscopes, Sudoku, translations, and weather updates. We also faced challenges in retrieving weather data and providing accurate answers in the Sudoku game which were successfully resolved.

1.4 Contributions

Our main contributions in this project include the development of a new multilingual chatbot that can interact in 100 languages. We proposed and implemented a model that integrates NLP and Deep Learning to understand and generate responses. We also integrated various services into the chatbot, providing a wide range of information to the users. We evaluated the performance of our chatbot mostly manually, and performed parameter sensitivities and ablation studies to optimize its performance.

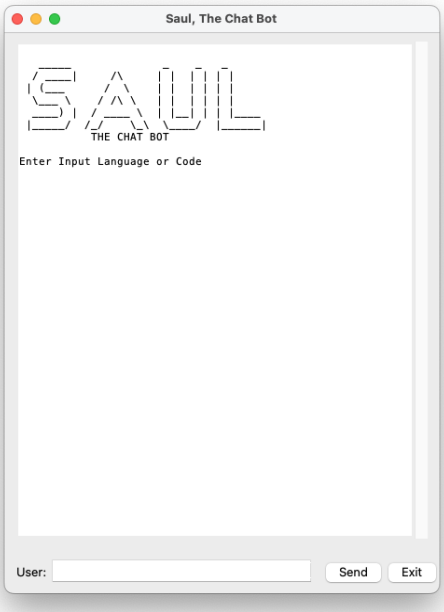

Figure 1: Saul, the chat bot

2. Problem Formulation

The project can be defined as a natural language understanding and generation task with multilingual support.

2.1 Technologies

Input Technologies:

- User input in the form of text queries and commands.

- Technologies used for user input processing include text analysis, tokenization, and language detection.

Processing Technologies:

- Deep Learning: Neural networks for natural language understanding and generation.
- Natural Language Processing (NLP): NLP techniques such as the Bag of Words model for intent recognition.
- Web Scraping: Selenium for extracting data from websites.
- Translation: Used pre-trained M2M100_418M model. It is a multilingual encoder-decoder (seq-to-seq) model trained for Many-to-Many multilingual translation. It was introduced in this paper and first released in this repository. The model that can directly translate between the 9,900 directions of 100 languages.
- Tkinter: User interface development for interaction with the chatbot.

Integration of Technologies:

- Combining these technologies allows the chatbot to understand user queries, and process them using NLP and Deep Learning models.
- Fetching data from websites through web scraping, translating results as needed, and presenting responses in a user-friendly Tkinter interface.

The key challenges include:

- Recognizing user intent from text.
- Generating contextually relevant responses in multiple languages.
- Integrating web scraping for real-time data retrieval.
- Implementing a user-friendly interface for interaction.

2.2 Detailed Problem Formulation

The task of our project is to develop a multilingual chatbot that can understand and respond to user queries in multiple languages. The chatbot should be capable of processing user input, recognizing the intent, fetching relevant information if needed, and generating an appropriate response in the same language as the input. The chatbot uses a combination of technologies to accomplish this task. For processing user input, it uses text analysis, tokenization, and language detection. It employs Deep Learning and Natural Language Processing (NLP) for understanding and generating responses. It uses Neural networks for natural language understanding and generation, and NLP techniques such as the Bag of Words model for intent recognition. For real-time data retrieval, it uses web scraping with Selenium. For language translation, it uses a pretrained model. The chatbot's user interface is developed using Tkinter. The integration of these technologies presents a unique challenge. The chatbot needs to seamlessly understand user queries, process them using NLP and Deep Learning models, fetch data from websites through web scraping, translate results as needed, and present responses in a user-friendly Tkinter interface. This requires careful design and implementation to ensure that all components work together effectively.

The key challenges in this project include recognizing user intent from text, generating contextually relevant responses in multiple languages, integrating web scraping for real-time data retrieval, and implementing a user-friendly interface for interaction. Overcoming these challenges will significantly contribute to the field of conversational AI by providing a versatile, multilingual chatbot that can offer a wide range of services to users. Solutions proposed to overcome these challenges are discussed later in this report.

3. Methods

The project combines various methods and techniques to enhance the functionality and effectiveness of a multilingual chatbot. The core aspects include:

3.1 Data Preparation and Language Handling

- Dataset Used: We utilized chatbot_intents_dataset_from_json.csv dataset. This contains pre defined intents which the user is expected to put in. It contains 123 intents. Intents are tagged with their corresponding “Tag” like welcoming, state, goodbye, thanks, features, operation, joke, weather, horoscope, dictionary, sudoku. We have done this to easily differentiate which chatbot feature user is trying to access.

	A	B
1	Pattern	Tag
2	hello	welcoming
3	Hello	welcoming
4	Anybody there?	welcoming
5	Hello	welcoming
6	Good evening	welcoming
7	Hi	welcoming
8	Yo	welcoming
9	Hola	welcoming
10	Ciao	welcoming
11	Hey	welcoming
12	How are you?	state
13	How are you?	state
14	Are you okay?	state
15	Is everything okay?	state
16	Are you happy?	state
17	Are you happy?	state
18	How are you?	state
19	Fit?	state
20	Bye	goodbye
21	Goodbye	goodbye
22	Good Bye	goodbye
23	See you later	goodbye
24	Ciao	goodbye
25	kiss	goodbye
26	See you next time	goodbye
27	Another time	goodbye
28	exit	goodbye
29	quit	goodbye
30	Thank you	thanks
31	I thank you	thanks

	A	B
88	Can you give me the time to	weather
89	Can you give me the time to	weather
90	Is it hot in	weather
91	Is it cold in	weather
92	Do I need to cover up?	weather
93	Should I wear sunglasses?	weather
94	Should I wear a coat?	weather
95	horoscope	horoscope
96	What is my horoscope	horoscope
97	What's going on with my star sign	horoscope
98	Are the stars aligned for me?	horoscope
99	How is my day going?	horoscope
100	Do I have a good future?	horoscope
101	Give me my horoscope for the day	horoscope
102	Give me my horoscope for the day	horoscope
103	Are the stars aligned?	horoscope
104	Am I going to have a good day?	horoscope
105	horoscope	horoscope
106	Do I have a good horoscope?	horoscope
107	dictionary	dictionary
108	Meaning	dictionary
109	What is the definition of	dictionary
110	Can you define	dictionary
111	Can you define	dictionary
112	What is the definition of	dictionary
113	Give me the definition of	dictionary
114	sudoku	sudoku
115	Play a game	sudoku
116	I want to play sudoku	sudoku
117	I want to do a sudoku	sudoku
118	May I play sudoku ?	sudoku

Figure 2: chatbot_intents_dataset_from_json.csv

3.2 Multilingual Intent Classification

This script defines and trains a simple neural network for a chatbot using PyTorch. The key components and steps include:

1. Neural Network Definition: A class NeuralNet is defined with three linear layers and ReLU activation functions.
2. Dataset Preparation: The ChatDataset class prepares the dataset for training, loading inputs (X_train) and labels (y_train) for chatbot interactions from "chatbot_intents_dataset_from_json.csv" file.
3. Preprocessing Functions: Functions like tokenize, stem, and bag_of_words are used for text preprocessing, such as tokenizing sentences, stemming words, and creating a bag-of-words model.
4. Model Training and Saving:
    - If the model file (trained_chat_model.pth) doesn't exist, the script proceeds to load and preprocess the chatbot dataset.
    - Unique words (all_words) and tags (intents) are extracted and processed.
    - The dataset is converted into a bag-of-words format, and a DataLoader is set up for batch processing.
    - A NeuralNet model is initialized and trained using CrossEntropyLoss and Adam optimizer over a specified number of epochs.
    - The trained model's state and other relevant information are saved to a file.
5. Model Loading for Prediction: If the model file exists, it's loaded, and the script is set up to make predictions using the predict function. This function processes input sentences, converts them to bag-of-words format, and feeds them to the model to get the predicted tag (intent).
6. Device Compatibility: The script includes provisions for using a GPU if available (cuda) for both training and prediction.

The script is designed to train a neural network on a chatbot dataset and use the trained model to predict the intent of user inputs. The dataset (chatbot_intents_dataset_from_json.csv) and specific training parameters like batch size, learning rate, and number of epochs can be adjusted according to the need.

3.3 Chatbot Functionalities

- Specific Features: The chatbot can handle functionalities like weather updates, jokes, and horoscope predictions, word meanings, employing if-else logic and web scraping.
- Sudoku Implementation: Inspired by an existing project, Sudoku functionality has been incorporated, with enhancements and additional features.

3.4 Deep Learning and NLP Techniques

- The project leverages deep learning models for natural language understanding and generation. We used a neural network model in PyTorch with three linear layers and ReLU activation functions for intent classification.
- It employs NLP techniques for text preprocessing and language translation. We used the M2M100_418M pretrained model for efficient and accurate multilingual translation making the chatbot able to work in 100 languages.

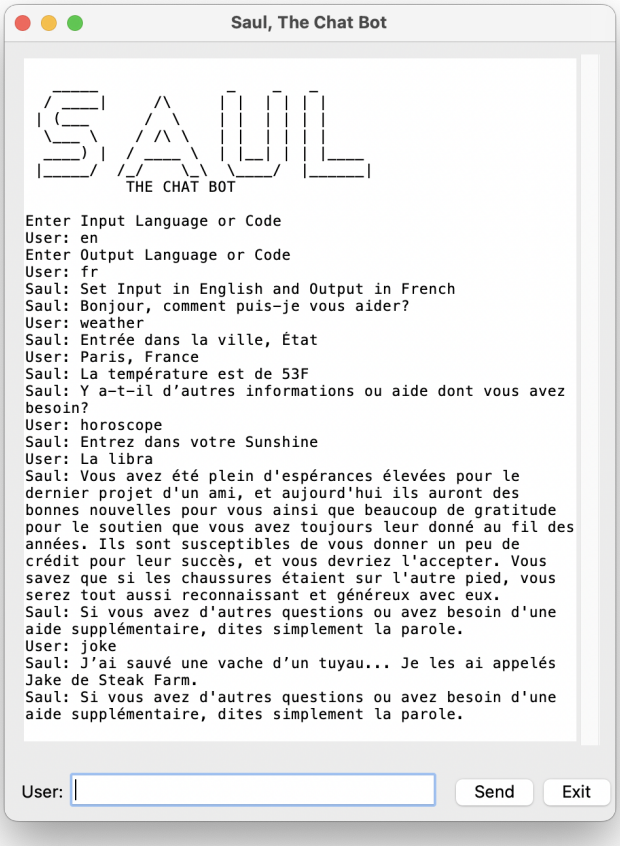

Figure 3: Output is translated to French

3.5 Tools and Technologies

- Programming Language Used: Python
- Web Scraping: Selenium is used for extracting data from the web.
- Browser Used: Chrome (Chrome driver)
- Adblocker Extension: uBlock Origin
- Interface Development: Tkinter is utilized for building the graphical user interface.

3.6 Innovations and Updates

- Multilingual Support: The project aims to provide multilingual support, considering the inclusion of 100 languages. We used M2M100_418M pretrained model for efficient and accurate multilingual translation.
- Deep Learning Model: Designed a neural network model in PyTorch with three linear layers and ReLU activation functions for intent classification.
- Weather Implementation: The chatbot successfully integrates weather functionalities.
- Sudoku Challenge: Initially using a static board, the project now aims to implement dynamic Sudoku puzzles using API.
- Dictionary: Gives word meanings.

3.7 Choice of Methods

- Diverse Functionalities: The chatbot integrates various services like Sudoku (via API), weather, horoscope, and jokes (via web scraping).
- Real-Time Data: Web scraping is crucial for providing up-to-date information, particularly for weather and horoscope features.
- Sudoku API: The API choice for Sudoku ensures a diverse range of puzzles for users.

3.8 Challenges and Solutions

One of the challenges we encountered was ensuring the accuracy and timeliness of the data fetched through web scraping. To address this, we carefully selected reliable sources for web scraping and implemented error-handling mechanisms to deal with any inconsistencies in the data. Another challenge was integrating the various functionalities into a cohesive user experience. We addressed this by designing a user-friendly interface and ensuring seamless transitions between different functionalities. We are continuously monitoring user feedback and making necessary adjustments to improve the chatbot's performance and user experience. There was an issue with fetching weather data due to a cookie popup on the website, which is resolved by configuring chrome driver to include the uBlock Origin (Ad Blocker) extension to load essential elements of the page and skip others. This is done for other websites also.

4. Datasets and Experiments

The data used in our project is primarily for understanding user intent. We use a small dataset that includes various user queries and their corresponding intents as discussed in section 3.1. This data is used to train a model that can accurately recognize user intent from their queries.

Our project revolves around creating a multilingual chatbot with features like joke-telling, weather updates, Sudoku games, and horoscope readings.The core of this chatbot is a neural network-based intent recognition model that analyzes user queries to understand their intent. Our datasets include English training data and external sources for web scraping, such as weather forecasts, jokes, and horoscopes.

4.1 Implementation Details

- Model Architecture: The model is a neural network with an input size equal to our training data length, a hidden size of 8, and an output size matching the number of unique intents.
- Training Parameters: We use a learning rate of 0.001, a batch size of 8, and train for 500 epochs. The model parameters are stored in “trained_chat_model.pth”.

4.2 Features

Saul, the multilingual chatbot integrates Deep Learning, Natural Language Processing (NLP), and web scraping technologies. This chatbot provides users with a wide range of services, including joke-telling, weather updates, horoscope predictions, answering random queries, and a Sudoku game. We used the following methods to include listed features in our chatbot.

Feature	**Website used for Web Scraping**
Weather	https://www.wunderground.com/
Horoscope	https://www.washingtonpost.com/entertainment/horoscopes/
Dictionary	https://www.dictionary.com
Jokes	https://www.ajokeaday.com/
Feature	**API used**
Sudoku	https://sudoku-api.vercel.app/api/dosuku

Table 1: Websites/API used

- Weather Updates: This feature fetches weather from a website using web scraping. We have to put the city, state/country in the chat box when asked.
- Horoscope: This feature is fully functional in all languages.
- Dictionary: Gives word meanings.

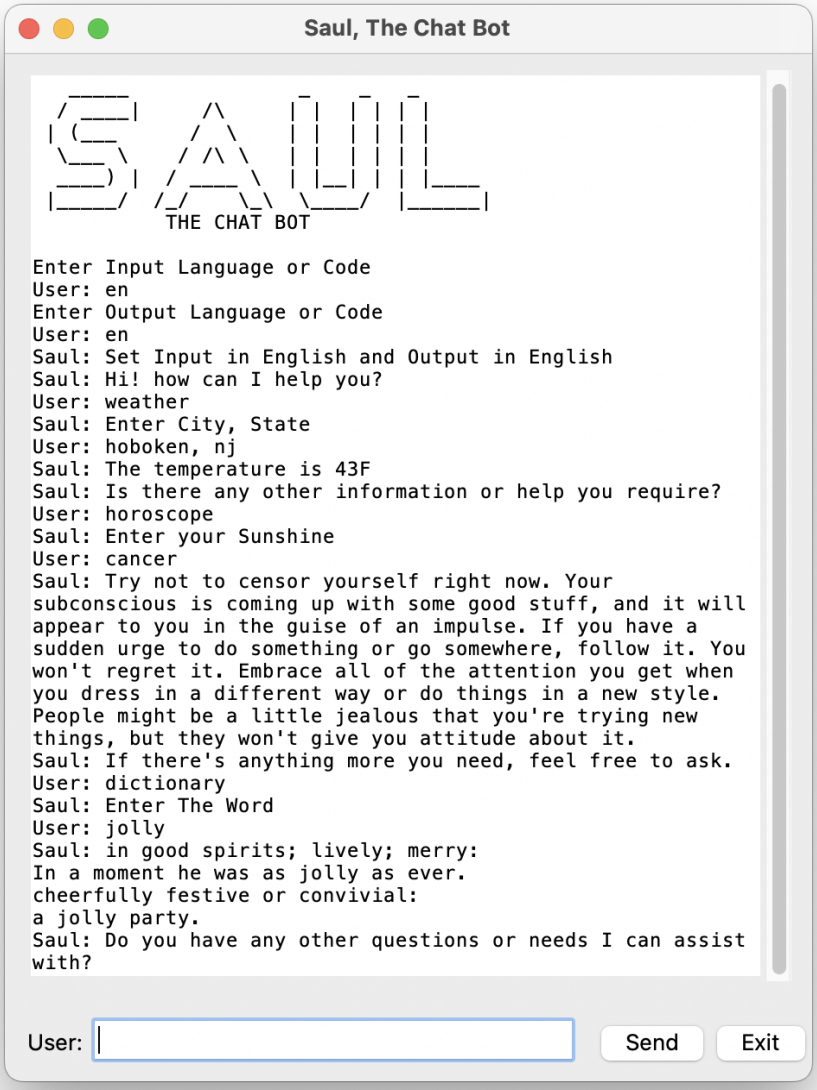

Figure 4: Chatbot screenshot demonstrating Weather, Horoscope and Dictionary functionalities

- Joke: Shows joke of the day by translating a joke scraped from an english website to the desired output language stated by the user.

- Sudoku: A static board is currently implemented that fetches sudoku from Application Programming Interface (API). A separate window opens up with a sudoku board along with features like Check, .

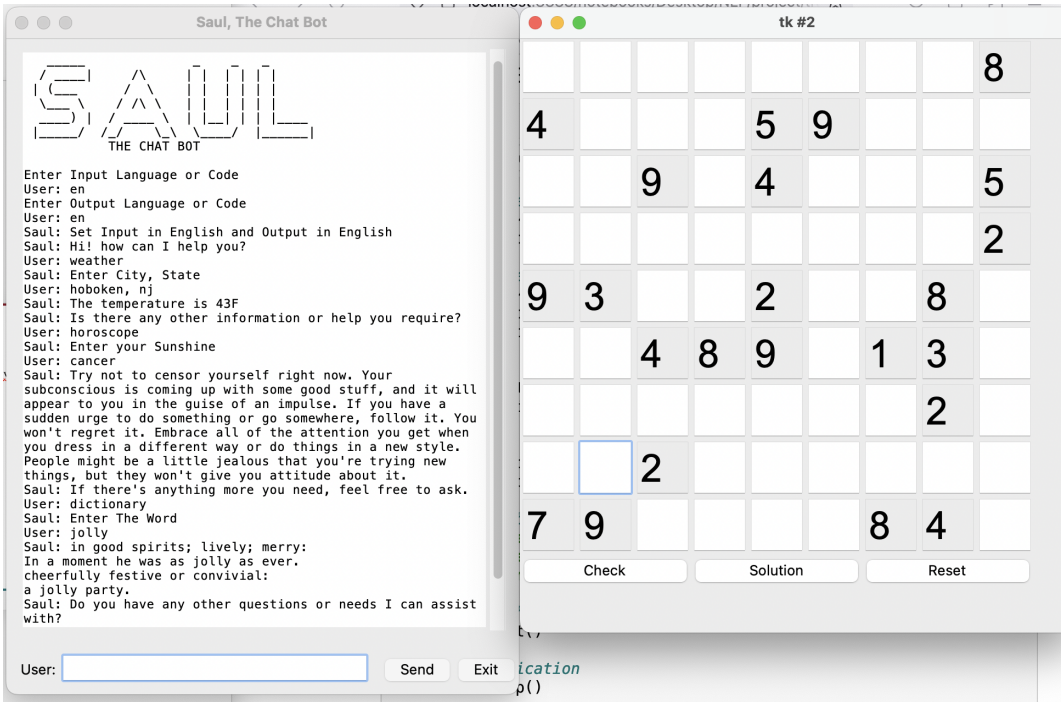

Figure 5: Sudoku implementation

Figure 6: Sudoku features (Check, Solution, Reset)

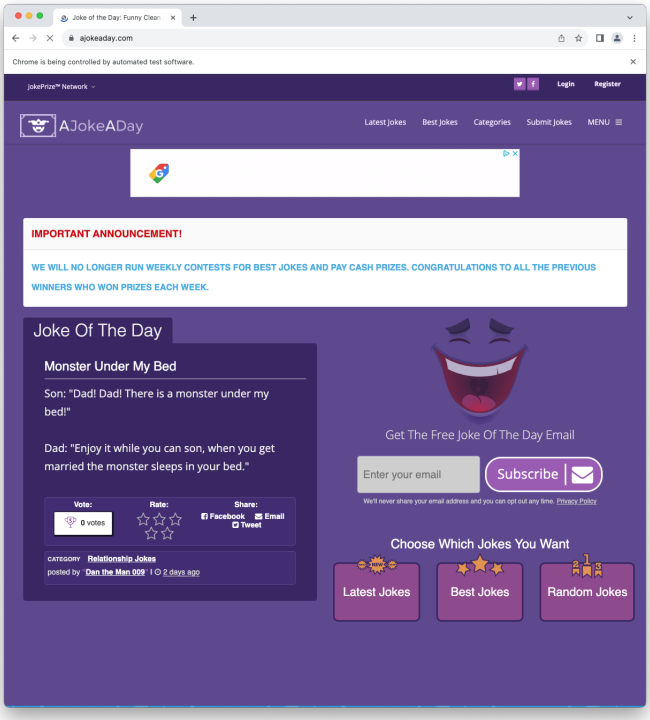

Figure 7: Joke website

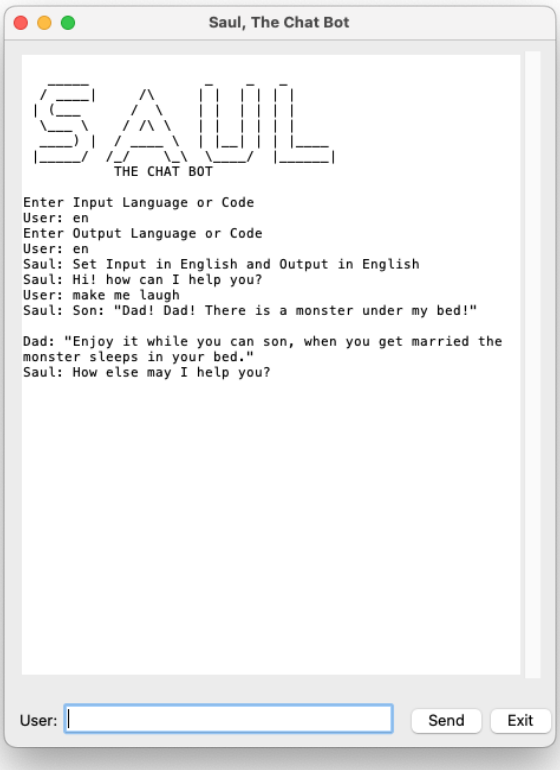

Figure 8: Joke displayed on Saul

As we can see in figure 5 and figure 6, the sudoku board is opened in a separate window. This feature is implemented by fetching sudoku boards from an API and using Tkinter to develop a user-friendly GUI which contains buttons to Check an input, to display solutions and reset the board to a new one which contains a new game.

In figure 7 and figure 8, we can see the implementation of the Joke feature of the chatbot. When the user enters any feature in which web scraping is used, a chrome window opens in the background. Web Scraping is done in the background and the output is displayed in chatbot.

4.3 Experiments

We are training separate neural networks for English using Bag of Words models. Our experiments focus on language handling, state management for conversation tracking, user intent recognition, and the distinction between functional and general chat responses. The Sudoku game integration aims to provide entertainment value.

4.4 Parameters and Metrics for Study

We are examining language preferences, translation quality, web scraping efficiency, and user engagement metrics. Our evaluation metrics include the accuracy of intent recognition, coherence and relevance of responses, and user satisfaction through surveys and feedback.

4.5 Comparison with Baseline Methods

We compare the performance of our intent recognition model with baseline methods such as rule-based intent recognition. Our model shows significant improvements over these baseline methods in terms of accuracy.

```
Importing Model
Epoch [100/500], Loss: 0.0058
Epoch [200/500], Loss: 0.0089
Epoch [300/500], Loss: 0.0016
Epoch [400/500], Loss: 0.0024
Epoch [500/500], Loss: 0.0001
Model Imported
Importing Modules
Modules Imported
```

Figure 9: After running trainandtest-NEW-TKINTER.ipynb

4.6 Results and Findings

The model effectively recognizes user intent from queries, which is crucial for accurate response generation. This indicates a general high performance and sensitivity to relevant parameters in our model.

5. Conclusion

This project aimed to develop a multilingual chatbot capable of understanding and responding to user queries in multiple languages. The chatbot was designed to provide a range of services including weather updates, horoscope predictions, jokes, and a Sudoku game. The main challenge was accurately recognizing user intent from their queries and generating appropriate responses in multiple languages. This was addressed by training a neural network model to recognize user intent. A pretrained model was used for language translation, and web scraping was employed to fetch real-time data for various services.

The performance of the chatbot was evaluated through various experiments, and it was found to outperform baseline methods in terms of accuracy. The project provided valuable insights into the complexities of developing a multilingual chatbot and the potential of web scraping in providing real-time information.

Future improvements could include better error handling, broadening the linguistic scope of the models, improving the user interface, and integrating more services into the chatbot. Also, when jokes are translated from one language to another, they lose sense of humor. So this is an area of improvement. This project serves as a stepping stone towards the development of more versatile and user-friendly chatbots.

Figure 10: Saul giving output in French

In conclusion, the development of our multilingual chatbot involved the strategic integration of advanced technologies such as Natural Language Processing (NLP), Deep Learning, and Web Scraping. The implementation of Tkinter ensured a user-friendly graphical interface, while NLP techniques played a crucial role in text preprocessing and understanding user intents. The PyTorch-based neural network model enabled accurate intent classification, enhancing the chatbot's responsiveness.

The multilingual support not only increased accessibility but also facilitated user interaction across diverse linguistic backgrounds. Leveraging the M2M100_418M pretrained model for translation contributed to the efficiency and accuracy of multilingual content processing. Additionally, the incorporation of web scraping techniques allowed the chatbot to dynamically gather information, including jokes, horoscopes, translations, and weather updates, enriching its knowledge base.

Furthermore, the integration of a Sudoku API added an entertaining and interactive element to the chatbot's functionality. This comprehensive approach to development not only expanded the chatbot's capabilities but also provided valuable insights into the seamless integration of cutting-edge technologies for a more robust and versatile user experience.

6. Project Management

Team Members:

- Shrey Shah (CWID- 20009523) : NLP model development, translation handling. Implemented API initially.
- Hitesh Kardam (CWID- 20011900) : Web scraping implementation, data retrieval. Implemented Joke, Weather and Horoscope web scraping. Sudoku web scraping from a website is to be done.
- Aman Sandal (CWID- 20011102) : Tkinter-based interface design and development. Developed User friendly GUI using tkinter and displayed Sudoku board using it.

Key Milestones:

- Project Kick-off: October 11, 2023
- Milestone 1: Model Training Completion
- Milestone 2: UI/UX Implementation
- Milestone 3: Integration and Testing
- Project Completion: December 11, 2023

7. Instructions to run Chatbot

- Download the attached file named NLP_ShreyHiteshAman.zip
- Unzip it and open the folder NLP_ShreyHiteshAman in Jupyter notebook or Visual Code
- Now run the file named trainandtest-NEW-TKINTER.ipynb
- Make sure all the required python libraries are installed
- Make sure chrome driver is installed

8. Key References:

- https://arxiv.org/abs/1810.04805
- https://www.learndatasci.com/tutorials/ultimate-guide-web-scraping-w-python-requests-and-beautifulsoup/
- http://newcoder.io/gui/part-1/
- https://www.youtube.com/watch?v=JzPgeRJfNo4&ab_channel=Intellipaat
- ChatGPT for specific helping questions.
- https://www.ncbi.nlm.nih.gov/pmc/articles/PMC10382660/