A report on the Coursework 1- Chatbot implemented using Natural Language Processing, programmed using Python.

CHATBOT XENO: HAI COURSEWORK 1- NLP

Kushagra Anand
20340188
psxka6@nottingham.ac.uk
University of Nottingham

CHATBOT XENO: HAI COURSEWORK 1- NLP

1. INTRODUCTION

In the age of machines, where Artificial Intelligence is taking command of many crucial things, a chatbot, has become very common. Xeno, the chatbot, is a simple yet useful output of natural language processing in python. It is used as a meal booking bot for a restaurant. The idea of using this bot in this domain is to demonstrate how quickly and easily fast-food orders can be placed more simply and interactively. It follows a simple flow and answers questions related to restaurant ordering and some simple questions regarding itself. It has intent matching, identity management, small talk and game playing. It is made by using simple concepts of Lists, TensorFlow and convolution networks, for training, one hot encoding, if-else and while loops, and of course Natural Language Processing.

2. BACKGROUND

The chatbot tutorial for intent making and small talk (Tech with Tim, 2021) was helpful. Intent matching is implemented there which served as an inspiration and reference for model training and usage of .json file for intent training. The paper (Schlesinger, O'Hara and Taylor, 2018) acted as a motivation for me to implement name extraction and highlight a single bias in the chatbot more clearly where it is able to extract English names better than any other. This has been discussed with examples in the discussion section. To improve name extraction and how to tag them using POS, the article at medium (Crosson, 2021) was helpful at clearing my doubts and a better understanding of the concepts. Instead of just tagging all the names and using it, I am just using the first name entered by the user in the chat bot. Basic doubts and list operation syntax were cleared during my lab sessions and on GeeksForGeeks (Agrawal et al., 2021). These resources along with my lecture notes served as an inspiration for me to make the chatbot.

3. PROPOSED SYSTEM

The chatbot is a combination of intent training, with text preprocessing like tokenization and stemming using snowball stemmer and then using a bag-of-words list for frequency check of the words present. When the user types a question or message in the chatbot, it looks through the string and sees the keywords and matches it with the intent tags and tries to reply with the closest possible response. The chatbot can cope up with dialogue flow, is interactive and follows the following Intent matching, identity management, small talk, one aspect of the transaction, which is food ordering, and has game playing implemented as well.

Generally, the accuracy of the model is good, around 97% as seen on the training metrics but it has been tested as we will see to it, in more detail, in the next section. For now, let us focus on the three key points for the chatbot.

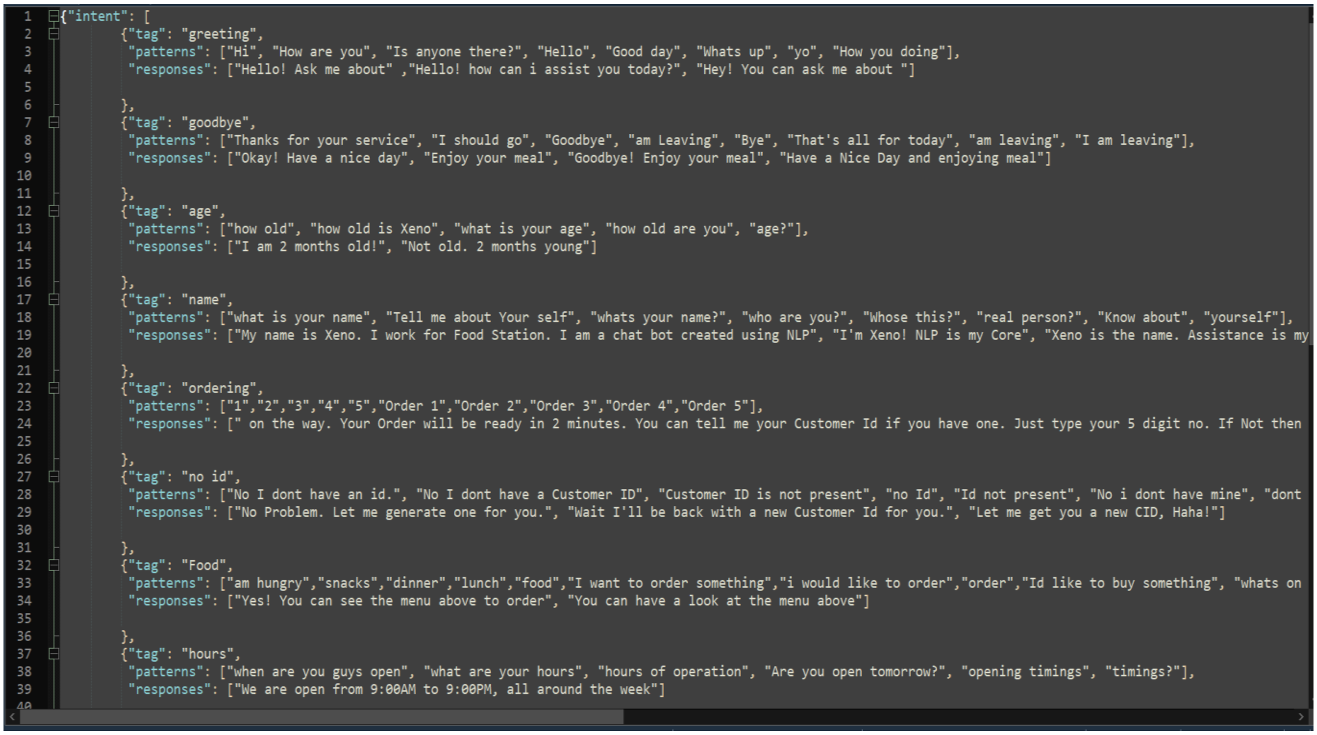
3.1 Functionality

The chat bot's function is very simple. It greets the customer and takes orders from them, which is a transaction-based action. It tries to demonstrate how convenient it can be to use a more interactive AI-based machine for a quick grab and go predefined options. It can also take note of the name and can use it to maintain logs as well which is an example of identity management. Displaying the food menu in a tabular format is also present. One can also play a game of Atlas, where the bit gives you a random letter and the user has to say a city's or country's name starting with that letter and the end letter of it should be the starting letter for next and so on. This takes care of game playing.

3.2 Originality

I have tried to make the flow of the chatbot simple yet as effective as possible. The intent file is used in almost every conversation of the bot as it is used to branch to other segments for a more free-flowing architecture of the system. It has an Atlas game that the user can play while waiting for the order. Combined with a good database of a restaurant, it can serve the purpose of a quick and easy food ordering bot which could remember the names of its customers and display menu and other data (if required) in tabular form. It is a mixture of rule-based and convolution networks combined with intent matching, following the basic principles of NLP.

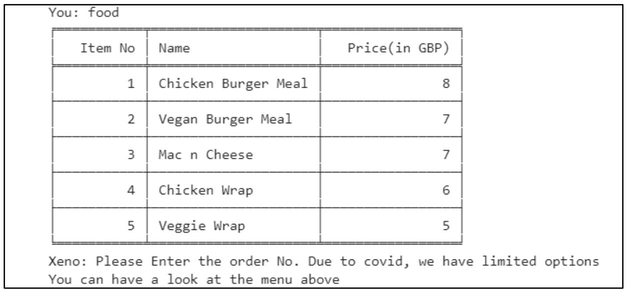
3.3 Implementation



```
{"intent": [
        {"tag": "greeting",
         "patterns": ["Hi", "How are you", "Is anyone there?", "Hello", "Good day", "Whats up", "yo", "How you doing"],
         "responses": ["Hello! Ask me about" ,"Hello! how can i assist you today?", "Hey! You can ask me about "]

        },
        {"tag": "goodbye",
         "patterns": ["Thanks for your service", "I should go", "Goodbye", "am Leaving", "Bye", "That's all for today", "am leaving", "I am leaving"],
         "responses": ["Okay! Have a nice day", "Enjoy your meal", "Goodbye! Enjoy your meal", "Have a Nice Day and enjoying meal"]

        },
        {"tag": "age",
         "patterns": ["how old", "how old is Xeno", "what is your age", "how old are you", "age?"],
         "responses": ["I am 2 months old!", "Not old. 2 months young"]

        },
        {"tag": "name",
         "patterns": ["what is your name", "Tell me about Your self", "whats your name?", "who are you?", "Whose this?", "real person?", "Know about", "yourself"],
         "responses": ["My name is Xeno. I work for Food Station. I am a chat bot created using NLP", "I'm Xeno! NLP is my Core", "Xeno is the name. Assistance is my

        },
        {"tag": "ordering",
         "patterns": ["1","2","3","4","5","Order 1","Order 2","Order 3","Order 4","Order 5"],
         "responses": [" on the way. Your Order will be ready in 2 minutes. You can tell me your Customer Id if you have one. Just type your 5 digit no. If Not then

        },
        {"tag": "no id",
         "patterns": ["No I dont have an id.", "No I dont have a Customer ID", "Customer ID is not present", "no Id", "Id not present", "No i dont have mine", "dont
         "responses": ["No Problem. Let me generate one for you.", "Wait I'll be back with a new Customer Id for you.", "Let me get you a new CID, Haha!"]

        },
        {"tag": "Food",
         "patterns": ["am hungry","snacks","dinner","lunch","food","I want to order something","i would like to order","order","Id like to buy something", "whats on
         "responses": ["Yes! You can see the menu above to order", "You can have a look at the menu above"]

        },
        {"tag": "hours",
         "patterns": ["when are you guys open", "what are your hours", "hours of operation", "Are you open tomorrow?", "opening timings", "timings?"],
         "responses": ["We are open from 9:00AM to 9:00PM, all around the week"]
```

Fig 1 Intent File snippet

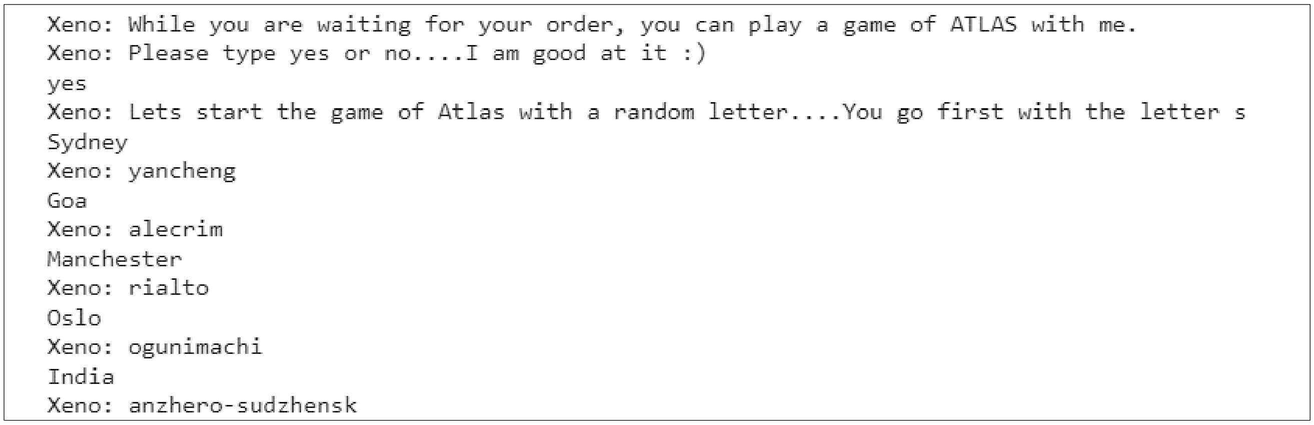
I created an intent file with different categories like greetings, opening hours, menu, etc. This json file is pre-processed using tokenization and stemming. For stemming snowball stemmer is used. After that, I used one hot encoding to see what all words are there and used a list (bag of words). After all the pre-processing is done, the model training comes into play. Since each training input is of the same length that's why the first layer's shape depends on the training list's first string. It is then followed by 3 hidden layers followed by an output layer. It is trained using the tflean library. It will help the bot to give better responses by picking up the keywords and responding to the questions rather than a simple rule-based approach. After the training and one hot encoding is done, the chatbot is ready for question answering and related small talk. It asks the name of the user and can extract the names with ease and uses the first name in the list of names to address the person concerned. It is done by using part of speech tagging (POS) which is tagging each word with grammatical labels. This is done only after tokenising the array or list of sentences that the user inputs at the first interaction where they are asked to enter their name. After POS is complete, chunking is done, which is to group and label multi-token sequences. Then I just had to separate the tag "PERSON" from the rest and then split it by space and store it for the chatbot to use it. After the user enters the name the flow transfers to data trained on the intent and one can begin chatting with the bot for ordering food, greetings, small talks like 'who are you?', 'what are your opening hours?' etc. For displaying the food menu, tabulate is used to convert the list in tabular form and this is being used by the chatbot to display the data in a tabular format for simple menu options for the customer to choose from.



```
You: food
 Item No | Name              | Price(in GBP)
       1 | Chicken Burger Meal |           8
       2 | Vegan Burger Meal   |           7
       3 | Mac n Cheese        |           7
       4 | Chicken Wrap        |           6
       5 | Veggie Wrap         |           5
Xeno: Please Enter the order No. Due to covid, we have limited options
You can have a look at the menu above
```

Fig 2: Tabular menu

When the user has ordered the food and is waiting, one can play the game of ATLAS where the bot gives a random alphabet (except x or q) to start the game. So, if it gives the letter ‘a’ then the user must name a city or country starting with ‘a’ and then the bot answers a random place with the last letter from the user’s input and so on. This is implemented using lists and normal text processing and string operations.



```
Xeno: While you are waiting for your order, you can play a game of ATLAS with me.
Xeno: Please type yes or no....I am good at it :)
yes
Xeno: Lets start the game of Atlas with a random letter....You go first with the letter s
Sydney
Xeno: yancheng
Goa
Xeno: alecrim
Manchester
Xeno: rialto
Oslo
Xeno: ogunimachi
India
Xeno: anzhero-sudzhensk
```

Fig 3: Chatbot playing Atlas with user

4. EVALUATION

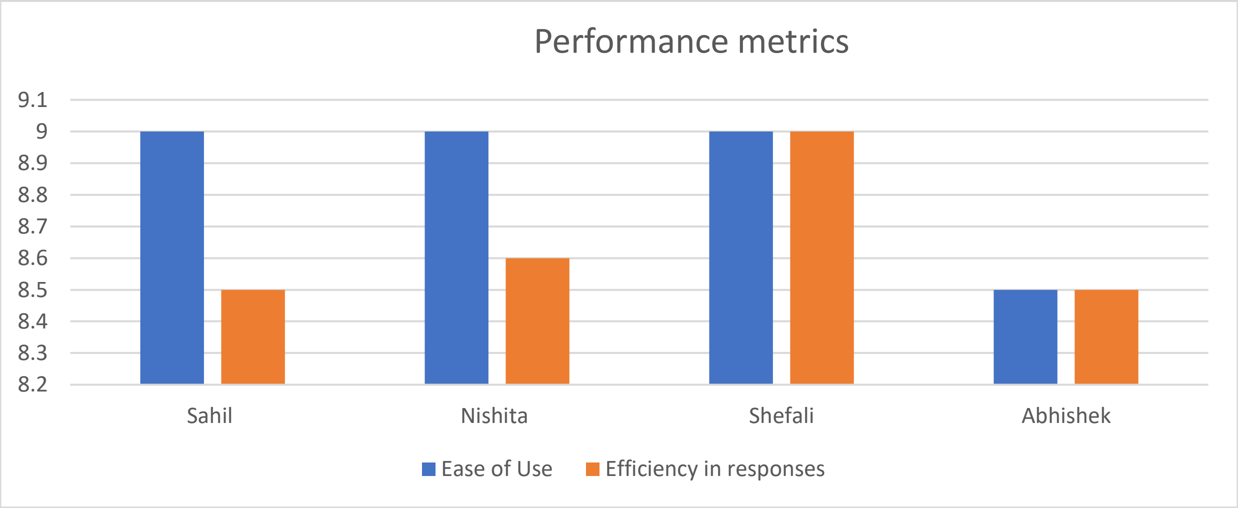


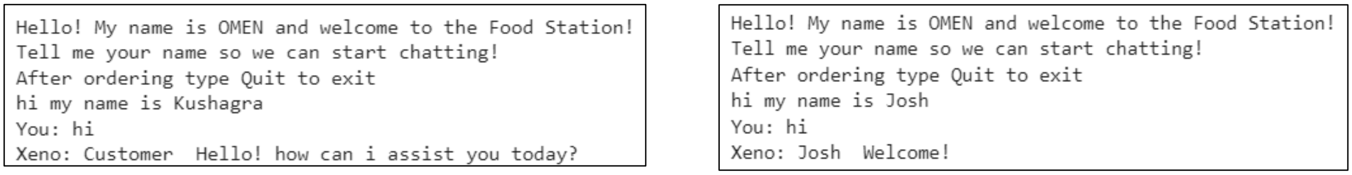
Fig 4: Comparison between Ease of Use and Efficiency in response amongst different users.

The chat bot implemented, is my first high level project in NLP. To evaluate how this bot performs, I asked my classmates and some known people doing master's in computer science or related field to test my chatbot and asked them to give a point based on the useability and overall effectiveness of the model, considering this as my first project in the domain of NLP. Each person was informed about the functionality of the chat bot and was asked to have a go at it. As displayed above, it got good score for ease of use and decent response for it efficiency. This User testing helped in evaluation of the model's performance and get an idea of how well the bot is performing.

5. DISCUSSION

After my own analysis and evaluation from my peers, I can conclude that this chatbot handles general food ordering queries in a decent manner. Handling ordering of food and letting customers play game while they wait for their order is an interactive way to keep the customers engaged. If my system were to be deployed to the real world, given a good database for the restaurant's menu, it could perform decent in the real life scenarios. Ideally customers can just come in, interact with the bot, give their order and then just wait for their order or play a game of atlas.

While there are no racial biases, after extracting the names using POS and chunking, Indian names are not recognised by the Bot as name and same goes with majority of Chinese names.



```
Hello! My name is OMEN and welcome to the Food Station!
Tell me your name so we can start chatting!
After ordering type Quit to exit
hi my name is Kushagra
You: hi
Xeno: Customer  Hello! how can i assist you today?
```

(b) Chatbot is unable to recognize Indian name

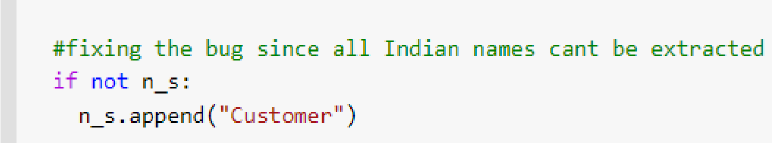
```
Hello! My name is OMEN and welcome to the Food Station!
Tell me your name so we can start chatting!
After ordering type Quit to exit
hi my name is Josh
You: hi
Xeno: Josh  Welcome!
```

(a) Chatbot recognizes English name

Fig 5: Chatbot Name extraction

This is the bias which the bot is having, and possible reason for this might be less diverse dataset for POS.

To avoid any errors, the bot is addressing the user as Customer if its unable to extract names in the given string.



```
#fixing the bug since all Indian names cant be extracted
if not n_s:
  n_s.append("Customer")
```

Fig 6: Code snippet for handling failed name extraction

5.1 Ethics

While speaking about the ethics of the chatbot, it can be said that it has a transparent nature where all the information is displayed when required and since it is not asking for any personal details like passwords, or card details, it can be said that it cannot use customer's data elsewhere except for the name which can just be used to maintain logs.

5.2 Results and Impacts

I would say that the responses from the chatbot are good, and it is easy to use except for some places where one might have to type a response twice. That's a minor bug and to improve on this, more programming practice must be done and more experience in NLP must be obtained. Xeno will create a positive impact when one is in a hurry and no modifications are required in the order. While not able to modify the order at the moment might send negative feedback.

6. CONCLUSION

After attending the lectures on Human AI Interaction which covered the topic Natural Language Processing, which included thermotical and practical knowledge, I was able to design this chat bot. Having no prior experience in NLP, I learned how to properly pre-process the text to make it suitable for further actions like model training, input for chat bots, speech tagging, etc. I used my knowledge of tokenisation, stemming, compared different stemmers like lancaster, snowball. I also used Speech tagging to grammatically find out the parts of speech of then sentences and used it to my advantage to extract names.

Overall, I would say the Xeno, the chatbot, is performing well enough for general queries related to restaurant fast food ordering, is able to interact in an average manner and let the waiting time for the food ordered, to be transformed into a fun game of atlas with the bot.

All in all, I learned the core concepts of NLP, got to know how to do text processing and improved my coding skills as well with this course work, which I can add to my CV.

7. REFERENCES

1. Agrawal, V., Aggarwal, N., Kumar, R., Arora, S., Singh, A. and Poonam, K., 2021. Python Lists - GeeksforGeeks. [online] GeeksforGeeks. Available at: <https://www.geeksforgeeks.org/python-list/> [Accessed 24 November 2021].
2. Crosson, A., 2021. Extracting names, emails and phone numbers. [online] Medium. Available at: <https://medium.com/@acrosson/extracting-names-emails-and-phone-numbers-5d576354baa> [Accessed 24 November 2021].
3. Schlesinger, A., O'Hara, K. and Taylor, A., 2018. Let's Talk About Race: Identity, Chatbots, and AI. CHI '18: Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems, Paper No.: 315, pp.Pages 1–14.
4. Techwithtim.net. 2021. Tech with tim. [online] Available at: <https://www.techwithtim.net/tutorials/ai-chatbot/part-1/> [Accessed 24 November 2021].