

Coursework 1: Multifunctional Chatbot

COMP3074

word count excluding references - 2175

Introduction

Chatbots is a concept that has been growing exponentially over the past few years within industry such as sports and social media. Functionalities have increased drastically, and they can be time and cost effective if used right. For this coursework I have been tasked to develop an interactive Natural Language Processing (NLP) system in the form of a chatbot, once handed the brief I choose to make my chatbot multifunctional and include features such as: intent matching, identity management, small talk and a question and answering element. My motivation for this is so I could explore all avenues of this coursework brief which I believed made the coding processes a lot more engaging.

Background Research

At first, I conducted some thorough research in order to grasp some insight into chatbots and gain inspiration on what tools and functionalities I could use. Furthermore, I investigated some of the history of chatbots to view the level of development that has happened from then till now. "An AI chatbot is a piece of software that interacts with a human through written language." [1]. The first ever chatbot was developed by a MIT professor named Joseph Weizenbaum the bots name was ELIZA, and this was created in the 1960s [2]. In this project my focus will be NLP however there are many different artificial methods that chatbots can use to communicate with users. For example, machine learning can be used for training methods.

In my code I focused on NLP my main tool used was the Natural Language Toolkit (NLTK) for python. This toolkit provided me with most tools needed to complete my chatbot. Formatively the "NLTK is a leading platform for building Python programs to work with human language data." [3]. Some useful features of the NLTK and what I made use of goes as follows: Tokenization, Lower case conversion, Stop Words removal, Stemming, Lemmatisation, Parse tree or syntax tree generation and POS tagging. All these elements allow me to pre-process the users input and the files that I intent to use.

Tokenization is when we break down into smaller units that are usually called tokens. **Stop Words removals** is almost like removing noise from an image it removes unnecessary word which don't help us with the processing element. **Stemming and lemmatization** both aim to find the root of a word, but they work in different ways. **Stemming** we commit the suffix and prefix to generate the 'stem' e.g., language -> languag and NLTK provides us with many kinds of stemmers such as PorterStemmer, SnowballStemmer and LancasterStemmer all which have their own positives and negatives. **Lemmatization** is similar but we extract the base form of the word here and call these a 'lemma' these words are available in the dictionary e.g., from naturally -> natural from the NLTK we can use the WordNet Lemmatizer for these tasks.

There have been many articles produced for the use of chatbots with in the sports industry with talks of them being used as promotional tools adding one-to-one interaction to any event. Furthermore, chatbots can serve as informational tools in term of tickets, timing and general facts. An article from chatbotslife.com [4], expresses the use of chatbots within the sport industry and

gave me the inspiration to create a football inspired chatbot. Especially with one of the biggest global sporting events, the world cup, taking place now.

Proposed System

My proposed system is a multifunction AI chatbot that is cable of the following features: intent matching, identity management, small talk, informational retrieval and questions and answering. The intent matching allows my chatbot to have so many functions and would allow me to easily add features if necessary.

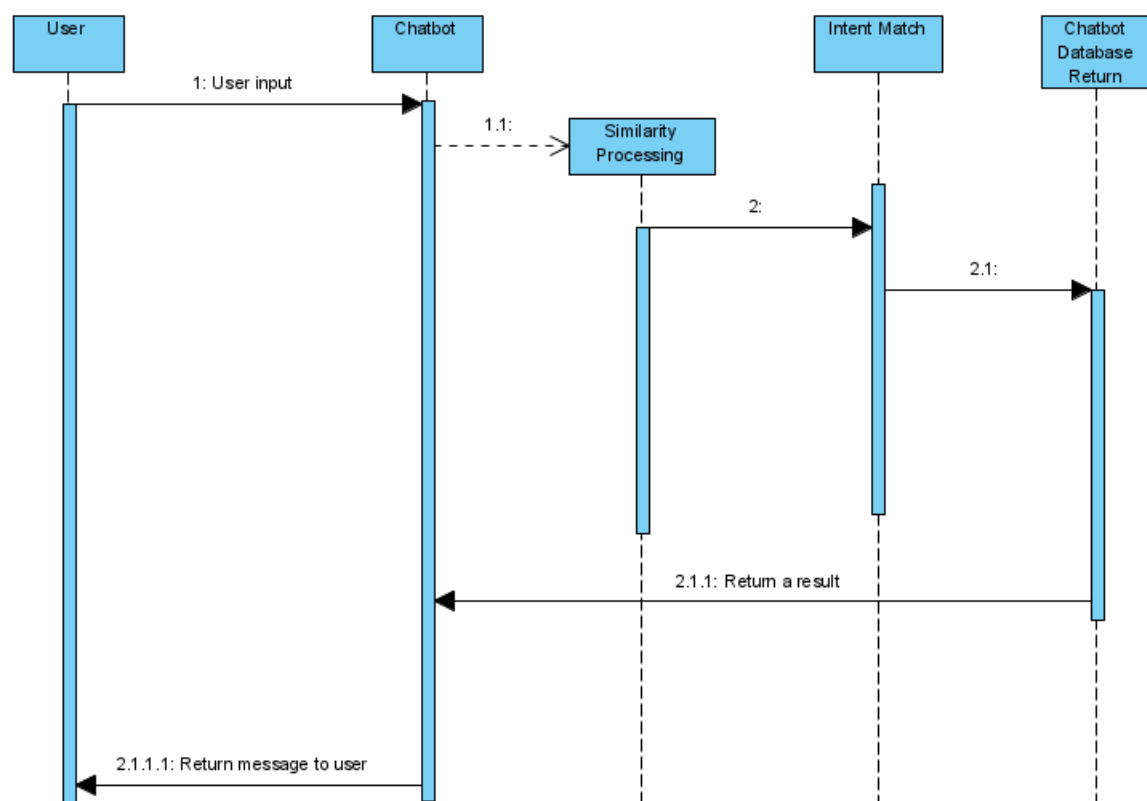


Figure 1: Simple Sequence Diagram representing the life of my AI Chabot

Pre-processing and Intent Matching

Using a method called pattern matching, I designed my chatbot to match the user input with an intent, all my patterns and intents were stored in a excel csv file for easy usability for me, which I then converted over to json file for processing ease. Before anything I had to begin some **pre-processing**, this is of the text and the user input. Step one was to remove any English stop words and punctuation from the text using the NLTK and the regular expression tokenizer for the punctuation. Rather than using stemming I opted for lemmatization from the NLTK's library this allowed the text words the change to become their base words which would make my chatbot have the chance to pattern match with ease. I prefer creating "lemmas" rather than "stems" as lemmas are in the dictionary so words will never come out misspelt or have the wrong meaning. Due to my database size the time efficiency of lemmatisation was not an issue that I faced. However, if I chose to increase my dataset stemming is an option that I could explore. I repeated this method for my user input and then moved onto my comparison function.

From the text I made a vector to compare the similarities at first, I used a count vectoriser but ran into issues at the conversion stage. After deeper research I opted for the TfidfVectorizer as I could avoid the steps of conversion and compute my term frequency-inverse document frequency (TF-IDF) weighting using the Scikit-learn library. This allowed me to find out the number of times a word occurs and compare them using NumPy and cosine_similarity functions to work out a maximum value and find a match. I then played around with a Threshold where I found 0.8-0.9 to be my sweet spot for comparison, I would select this intent if under this threshold the bot would question the intent and give a suggestion.

Identity Management

This section of my AI chatbot can allow the user to: change his name, ask their name or ask the bot's name this can be shown in figure 2. This allows the user to have a much more personal connection with the chatbot and they can change their name if they would ever like to. My system uses tokenisation and stop word removal to find the user's name.

```
Messi: Hi!, I am a multifunctional bot here to assist you :)
User: What should I call you?
User: my name is deonte
Messi: Hi deonte, I'm Messi but some call me the GOAT, let's get started if you need help or want to exit let me know!.
deonte: what is my name
Messi: Don't be silly we go way back deonte
deonte: what is your name
Messi: You can call me Messi
deonte: can i change my name
Messi: Were you asking about changeName yes/no: yes
Messi: What should I call you?
User: john
Messi: Okay I will now call you john
john: thanks
Messi: Your most welcome :D
john:
```

Figure 2: Example of my Identity Management working

Small Talk

My small talk section uses the same methods as my identity management. It uses pattern matching to find the user intent of small talk. Afterwards the response generated is a random selection of answers in an array. This gives my chatbot a natural following conversation feel and can be shown in figure 3 and 4 below.

```
deonte: hello
Messi: Heyy, how may I assist you
deonte: thanks
Messi: Anytime!
```

```
deonte: what time is it
Messi: The time now is 19:42
deonte: _
```

Figure 3 & 4: Example of my chatbot having a flowing conversation

Question Answer/ Information Retrieval

During my question and answering section I used similar pre-processing method to my intent matching. However, I adjusted the columns in which my processor refers to. This was changed to

refer to my question-and-answer dataset. I then removed stop words in the questions as these did not add an extra meaning.

The special part of my pattern matching is that if the questions is between a 0.4 and 0.8 threshold, I make my AI chatbot give a suggestion on what the user could be asking which is something I believe not many chatbots do from my own research.

Furthermore, during my football quiz if you are to get any of the questions wrong then you can retrieve the write answers from my chatbot as all the correct answers are in my dataset. An example of this can be shown in my video.

```
deonte: who is that smart man in the wheelchair?
Messi: Is your Question Stephen Hawking in popular culture related? yes/no: yes
Messi: Professor Stephen Hawking , known for being a theoretical physicist , has appeared in many works of popular culture .
deonte: what about kirk douglas
Messi: Is your Question Kirk Douglas related? yes/no: yes
Messi: Kirk Douglas (born Issur Danielovitch, ; December 9, 1916) is an American stage and film actor, film producer and author.
deonte: ■
```

Figure 5: Example of my question and answering system working

Evaluation

I started my evaluation with the main aspect of my code which is the question-answering system. I analysed my chosen threshold to see if any of my given tests would pass or fail without adjustments. Below I have found that all my chosen cases had passed or partially passed with the expected outcome. Despite this, many of my cases partially passed. This was because my similarity for exact answers was too high. Therefore, I lowered the maximum similarity to 0.75 and then I change my suggestion range to 0.4 – 0.7 this fixed my issue and did not give suggestions. However, I did not need to change my threshold because of my suggestion feature which allows error handling and prevents the user from getting the wrong information. Furthermore, this could cause me the issue of suggesting the wrong thing with future testing. Here below are the questions I used for testing all the user inputs are slightly different from the dataset questions.

Case	User Input	Expected Output	Pass/Partially Pass/ Fail	Issue	Final Result
1	Who is that smart wheelchair guy?	"Professor Stephen Hawking, known for being a theoretical physicist, has appeared in many works of popular culture ."			
2	Do you know who Kirk Douglas is?	"Kirk Douglas (born Issur Danielovitch,; December 9, 1916) is an American stage and film actor, film producer and author."		Gives correct suggestion instead of instant answer	
3	What is my name?	Reply with the users entered name and a randomised message			
4	Give me some information on the owls	"Owls are a group of birds that belong to the order Strigiformes, constituting 200 extant bird of prey species."		Gives correct suggestion instead of instant answer	

5	Do pineapples grow on trees	"Pineapples may be cultivated from a crown cutting of the fruit, possibly flowering in 20-24 months and fruiting in the following six months."		Gives correct suggestion instead of instant answer	
6	Are there many books in the bible?	"Christian Bibles range from the sixty-six books of the Protestant canon to the eighty-one books of the Ethiopian Orthodox Tewahedo Church canon."		Gives correct suggestion instead of instant answer	
7	What are Cyberstalkers?	"Cyberstalking is the use of the Internet or other electronic means to stalk or harass an individual, a group of individuals, or an organization."		Gives correct suggestion instead of instant answer	
8	How do kimberlite pipes form?	"Volcanic pipes are relatively rare."			
9	What does Ronaldo do when celebrating his goals?	""Suuuuuuii!" (Spanish for 'Yes!')"		Gives correct suggestion instead of instant answer	
10	How does a water pump work?	"Pumps operate by some mechanism (typically reciprocating or rotary) and consume energy to perform mechanical work by moving the fluid."			

Following this my case testing I also made a few general assumptions of what the chatbot should be able to provide for each case:

1. User would like to change his name, ask the time and play a short game.
2. User would like to know what the weather is like and then ask to try a football quiz and receive feedback.
3. User would like to greet the bot have some small talk and then ask to about "Kirq Douglas".
 - My chatbot does not user any method for typos which is something that I would implement in any future directions. Despite this, I found that my suggestive feature alleviates any of these issues for small cases.

Discussion & Conclusion

The main aspect of my chatbot was my similarity thresholding and this played a big part on my chatbots performance. Before I began my thorough user testing, I had problems making my chatbot recognise my user input without it being the exact same or extremely similar. But I also found that I couldn't making my maximum threshold too low as this was cause problems of the opposite effect.

Once I finish my evaluation and user testing, I found that a good suggestion range was "0.35-0.7" for my threshold and a good similarity for the answer was 0.75. Before I made these changes, I found that my bot would make the wrong suggestions and even sometimes give the wrong information for things the user did not ask.

Another issue that I found was user typos this is something that is common, and my chatbot does not have a way to compute this. Here in an example, we can see that if the trigger words are spelt correct and everything around it is spelt wrong then my bot can still compute answers from my suggestive functions. But without the trigger words spelt correctly my bot fails to understand the user so this is something that I would alter in the future.

```
john: let play a gme
Messi: Is your Question Big Bird related? yes/no: no
Messi: Sorry I am not sure what you mean :(
Messi: None
john: lts ply a game
Messi: Welcome to the number guessing game!
Messi: I am thinking of a number between 1 and 10. Can you guess what it is?
john:
```

Figure 6: Example of misspelt words

To conclude I would like to look at future steps for my chatbot I believe that currently my chatbot has a very engaging, fun experience and provides my user with multiple different functions. Such as playing a game, playing a quiz, asking the time, asking the weather or even asking random questions. However, taking my bot further I would like to incorporate more use of the internet and have my chatbot train itself in a way where the database can but updated on things that it does not know. Furthermore, a few smaller things that I would need to include is translation as my chatbot only communicates in English and a filter for any profanity so that my chatbot can be friendly for all ages as it is multifunctional so all ages should be able to use it.

References

- [1] "What is an AI chatbot?," CM, [Online]. Available: <https://www.cm.com/glossary/what-is-ai-chatbot/>.
- [2] "The History Of Chatbots – From ELIZA to ALEXA," ONLIM, 2022. [Online]. Available: <https://onlim.com/en/the-history-of-chatbots/#:~:text=The%20first%20chatbot%20ever%20was,It%20was%20called%20ELIZA..>
- [3] "Natural Language Toolkit," NLTK, [Online]. Available: <https://www.nltk.org/#natural-language-toolkit>. [Accessed 12 2022].
- [4] "Sport Chatbots-A Step Towards Smarter Event Management," chatbotislife, [Online]. Available: <https://chatbotislife.com/sport-chatbots-a-step-towards-smarter-event-management-c45d6cfc624d>. [Accessed 12 2022].