

Intelligent Systems: Mini-Project (ChatBot)



universidade
de aveiro

MP2

Master in Informatics Engineering
Pedro Iglésias (89318)
Pedro Silva (93011)
João Soares (93078)
Wei Ye (93442)
21/06/2022

Technologies and Tools I

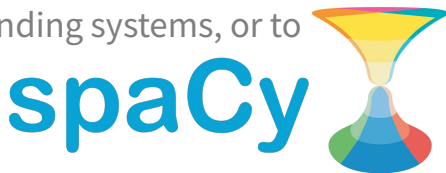
Search and experiment with different technologies and tools to use in the development of the conversational agent.

- **Python**
 - Easy language to use
 - Great experience/comprehension by the group members
- **Prolog**
 - In line with the objectives of the project - Designed to do natural language processing.
 - Familiar for the members of the group.
- **PySwip**
 - Enables SWI-Prolog in Python programs.
 - Provides (incomplete) SWI-Prolog foreign language interface, a utility class and a Pythonic interface.
 - Some experience by the group members.



Technologies and Tools II

- **OWLReady2**
 - Enables Owl language files to be loaded and used as python functions and classes.
 - It can be used as a database and as a brain for the AI.
 - Some experience by the group members.
- **Senticnet API**
 - Simple API to use SenticNet
 - SenticNet is an initiative conceived at the MIT Media Laboratory in 2010.
 - Concept-level sentiment analysis.
- **WordNet**
 - Large lexical database of English.
 - Nouns, verbs, adjectives, and adverbs.
 - Useful tool for computational linguistics and natural language processing.
- **Spacy**
 - Free and open-source library for Natural Language Processing (NLP).
 - Designed specifically for production use.
 - Can be used to build information extraction or natural language understanding systems, or to pre-process text for deep learning.



ChatBot Concept - Main Conversation Theme

- Using **English** Language
- Main Topic - **Marvel Cinematic Universe (MCU)**:
 - Movies, Characters, Storyline
 - Opinion of the user about the movies or characters



ChatBot Concept - Tamagotchi

The main idea is to create a chatbot around the virtual pet theme:

- Main focus Conversation:
 - Talking based on a theme and his mood
- Feeding:
 - He can get hungry, changing his mood, by feeding him he can learn new foods.
- Sleeping:
 - He can go to sleep and you can wake him up early.
- Playing:
 - You can play a game with him.



ChatBot Concept - Tamagotchi: Feeding

The Chatbot can get hungry and you can feed him new or old foods:

- With a new food, he learns its name and has a chance of liking or disliking the food which changes his mood.
- With old foods, giving him a food he dislikes will make him angry or grumpy, if you give him a food he likes or loves he will be happy.
- Or the user can just starve his pet making him very angry.



ChatBot Concept - Tamagotchi: Sleeping

If the Chatbot gets bored because you are not talking to him or doing anything with him he may go to sleep:

- If he is sleeping and the user wakes him up early, there is a chance that he will get angry and then he won't talk to you for a while.
- If he wakes up on his own, he wakes up not angry.



ChatBot Concept - Tamagotchi: Playing

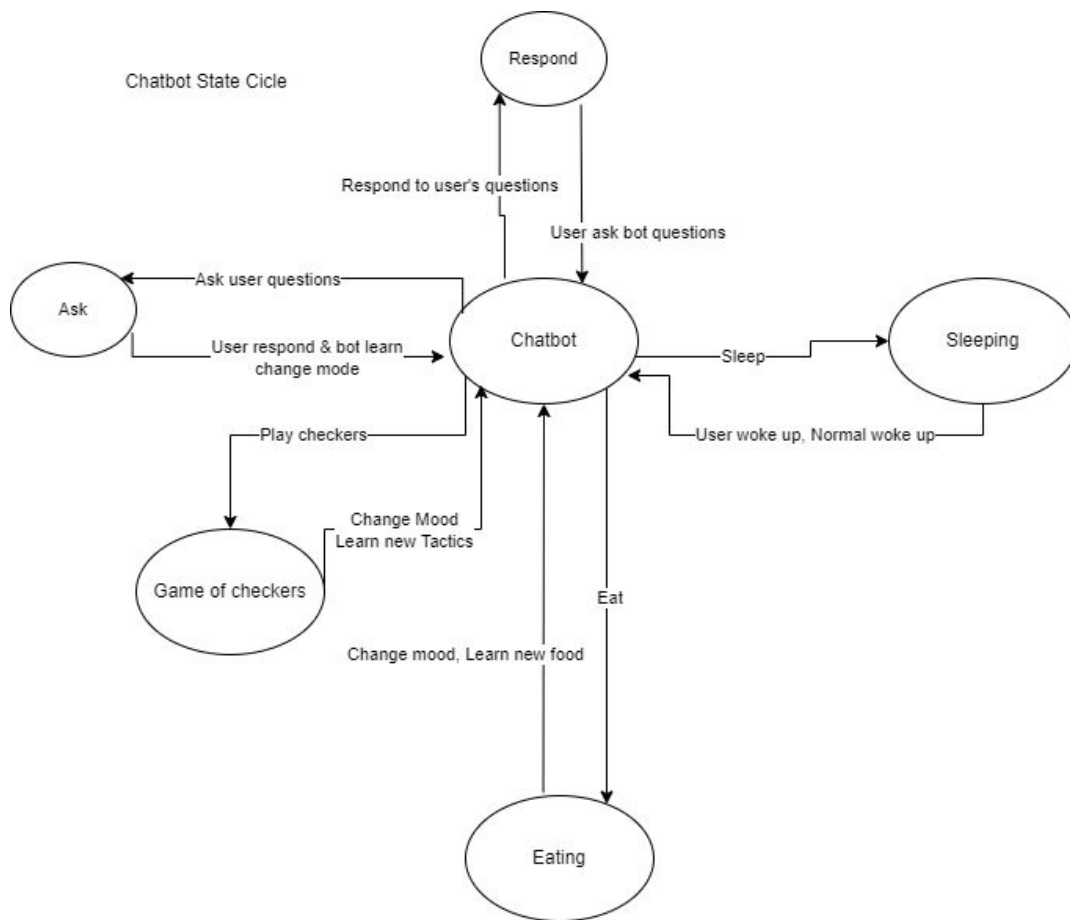
The ChatBot may get bored, the user can play a game (Checkers):

- If you play and the bot wins, he gets extremely happy.
- If you play and you win, the Chatbot gets happy or a bit annoyed.
- If you don't play, he gets bored or grumpy.



Design/Structure I

- ChatBot States



Implementation - Processing User Sentences

- **Spacy**, to identify the key words of the answers of the user to properly understand the information and the Chatbot give a corresponding answer.
- Arranged data to a format supported by Spacy and generated a model in a process that involves generating different sentences that include characters, movies and follow certain templates/syntax that the ChatBot supports.
- To process the user sentence, we first check the type of sentence and developed grammatics that are used in the main Python program with **PySwip**, for the possible sentences that the user can give, considering the context and topic.
- For each type of sentence, we used the model to identify the key words to better understand the context of the sentence and processed it with the proper processing of the sentence



Implementation - Answering User

- To give different answers to the user, we used **WordNet** lexical database taking advantage of its amount of verbs, nouns, adverbs and adjectives.
- For each conversation/interaction topic previously presented, we developed grammatics that are used in the main program with **PySwip**, for the possible answers that the ChatBot can give considering the context.
- In some of these topics/interactions, especially the main topic of conversation, the user may make questions that the ChatBot needs to consult its knowledge to answer, for that, the ChatBot knowledge is saved in the file *botMemory.owl* using **Owlready2**.
- When the ChatBot is initialized, it checks the file and loads the different types of knowledge (user name, characters, movies, etc...).



Implementation - Ability to Accumulate Information/Knowledge

- When user makes a question, the ChatBot checks its knowledge and if it knows the entity/context of the question produces the answer to it, else tries to understand the sentence and learns about the entity/context by different methods:
 - If the user makes a question about a movie/character, it asks the user for information about it and its opinion about it and at the same time accesses [Wiki MCU](#) to learn more information. In the end saves the new knowledge provided by the user and the results from the Web search.
 - If the ChatBot is fed with an unknown food to him, it will learn its name, form an opinion about it (liking or disliking) and then saves it.
 - If the ChatBot doesn't know the name of the user, it will question him and then save the name given in the answer.
- Because the new knowledge/information is saved in the file, this knowledge will remain learned even after closing the ChatBot and will be used in a next execution of the ChatBot.



Implementation - Tamagotchi Emotions I

- Mood is an important aspect of the ChatBot, as it influences how it responds to the user by influencing what actions the bot takes and what type of questions and answers it makes to the user, by way of using different grammars for questions and answers used.
- It uses **Senticnet** in order to analyze emotion from what it gets from the user which influences its own mood, it does this by analyzing all the emotions it finds in a sentence and choosing the most noticeable to be the new emotional state of the bot.
- Interacting with the user leads to the creation of events, represented in the form an enum, there are 3 actions that can happen, *likedFood*, *playAGame*, *dislikedFood*.
- These events are fed to the function *changeEmotionBasedOnAction* which in turns changes the ChatBot emotional state, but it can also change mood due waking up from sleep, depending on the state the ChatBot may go to sleep on its own or tell the user he is hungry.

Implementation - Tamagotchi Emotions II

- The bot can go through the following emotional states:
 - Satisfied
 - Calm
 - Bored
 - Angry
 - Sad
- When the bot changes emotion that change is printed to the screen so the user can react accordingly.

```
- i hate you  
{You see him sad }  
ChatBot: I am sorry i still didnt get that i dont understand could you please repeat  
- i love you  
{You see him happy }  
ChatBot: Come again ? i did not catch that would you mind repeating that
```

Implementation - Tamagotchi Interactions I

- **Feeding**

- Grammar to communicate to the user that it feels hungry, or if he likes/dislikes certain foods given to him. We also developed a grammar to interpret the sentences of the user regarding this interaction, such as giving new/old food to the ChatBot.
- As mentioned previously, giving foods to the ChatBot can affect its mood and he can learn new foods and has a chance of liking or disliking them which also affects its mood.
- The way the ChatBot starves is by not being fed during an amount of time.

- **Sleeping**

- ChatBot as the behaviour explained previously, using counter of the time passed without interaction.



Implementation - Tamagotchi Interactions II

- **Playing**

- To play Checkers with the ChatBot we developed a grammar to support the commands that the user is capable of doing in the game.
- Developed a class Checkers that provides and supports the methods to play the game and the way the ChatBot communicates with the user to play the game.
- As mentioned previously, playing the game and the end result will have an impact in the emotions of the ChatBot.



Demo

- Demo and Guide regarding the Main Conversation Theme and Demo and Guide regarding Tamagotchi Interactions.
- See more in the Report.