

UNIVERSITY AVEIRO



Mini-project - MP2

Intelligent Systems

Pedro Iglésias (89318)
Pedro Silva (93011)
João Soares (93078)
Wei Ye (93442)

Department of Electronics, Telecommunications and Informatics

2022

Contents

1	Introduction	1
2	Technologies and Tools	2
3	ChatBot Concept	3
3.1	Main Conversation Theme	3
3.2	Main Idea - Tamagotchi	3
3.2.1	Examples of Conversation	4
4	Design / Structure	6
5	Implementation	8
5.1	Processing User Sentences	8
5.2	Answering User	8
5.3	Ability to Accumulate Information/Knowledge	9
5.4	Tamagotchi Emotions	9
5.4.1	Happy	10
5.4.2	Satisfied	10
5.4.3	Calm	10
5.4.4	Bored	10
5.4.5	Angry	10
5.4.6	Sad	11
5.5	Tamagotchi Interactions	11
5.5.1	Feeding	11
5.5.2	Sleeping	11
5.5.3	Playing	11
6	Demo	12
6.1	Demo/Guide Main Conversation Theme	12
6.2	Demo/Guide Tamagotchi Interactions	13
	References	14

Chapter 1

Introduction

The use of conversational agents like chat bots provides an easy and natural way of getting information from a conversation indistinguishable from one had with a human being, considering this, movie franchises have become nowadays a huge source of information due to the existence of various connected stories spread around various movies, with the biggest example of this being the MCU, the Marvel Cinematic Universe.

With this project we have built a natural-sounding chat bot that is capable of answering questions about this movie franchise while also being capable of learning at runtime more information about it and the user's preferences concerning movies, it is also capable of sensing emotion in the responses given to him, and has is himself capable of feeling emotion which affects how it answers the user.

In order to make a finite state machine for the emotional state the bot could be in we decided to base ourselves around the concept of the portable virtual pet Tamagotchi where a small pet would have various needs and emotions and react to the user depending on these, in the same way, our agent is capable of going through various emotions which can be influenced by the user by doing activities like speaking to him, learning, feeding him or letting him sleep, these emotions will affect how the agent acts with the user just like a person would.

Chapter 2

Technologies and Tools

For this project, it was made an extensive search of different technologies and tools to use in the development of the conversational agent.

After reading and experimenting with some of the recommended works such as **Eliza Chatbot**[7][12], **Creating a Chat Bot**[10] and **ChatterBot**[9] we gathered some of the bases and concepts for the project, and decided that the best technologies to use in this work would be **Python**[13] as it is a relatively easy language, all elements of the group have a great comprehension of it and the use of **Prolog**[2] an acronym for 'Programmation en Logique' very in line for the objectives of the project, in other words, designed to do natural language processing, that is familiar for all members of the group (as is being learned in classes).

Owlready2[11], a python package that allows the creation, modification, and use of OWL ontologies in order to make an AI, using Python. It was chosen to take into account the experience of some group members.

For the connection between Python and Prolog we decided to use **PySwip**[14], which enables us to query SWI-Prolog in Python programs, using an (incomplete) SWI-Prolog foreign language interface, a utility class and a Pythonic interface.

For a list of words to use in the conversational agent we will chose words from the **WordNet**[15] lexical Database. It has a good amount of verbs, nouns, adverbs and adjectives for the project, which are grouped into sets and stored in different files.

For the Chatbot emotions we learned how to use **Senticnet**[3] through the API[4], that can be used to detect the various emotions on a word or a group of words, this helps will help when choosing how the Chatbot will respond and how the Chatbot will understand emotion considering the phrase that was given to him.

Spacy[5] is free and open-source library for Natural Language Processing (NLP) capable of information extraction, natural language understanding systems, or to pre-process text for deep learning. For the conversational agent it will be used to identify the key words of the answers of the user to then better understand the information and give the user the appropriate answer.

Chapter 3

ChatBot Concept

3.1 Main Conversation Theme

The main topic of conversation between the user and the conversational agent will be in English and related to The **Marvel Cinematic Universe**[1] (MCU) in which movies, characters, directors, and storylines will be themes as well as the opinion of the user about the movies or characters.

The agent will accumulate different information about the MCU, such as the opinion of the interlocutor about a movie or character.

The conversational agent will also “sense” negative and positive emotions from the interlocutor through specific words and sentences provided by him that then the agent will interpret.

3.2 Main Idea - Tamagotchi

The Chatbot will be developed around the idea of a virtual pet theme such as a Tamagotchi, "a keychain-sized virtual pet simulation game"[6], to test "your ability to take care of a small alien pet"[8], in a 2 small egg-shaped computer (Figure 3.1), that can fit into in a pocket.



Figure 3.1: Tradutional Tamagotchi

No matter the Tamagotchi, it was a needy little creature and required almost constant attention to keep it in good health, *needed to be fed, played with, nurtured, and disciplined*, just like a real pet.

ChatBot Concept

Functions and Interactions of the ChatBot influenced by the Tamagotchi will be:

- 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, he gets angry or grumpy if you give him a food he likes or loves he gets happy.
 - Or the user can just starve his pet making him very angry.
- 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.
- Playing - The ChatBot may get bored, the user can play a game (Checkers):
 - If you (user) 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.

3.2.1 Examples of Conversation

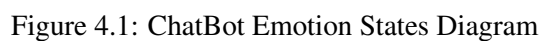
- What's your name?
 - Bot remembers this noun as user name
- What's your favorite movie?
 - [movie] -> Bot **will remember that**
- What's your favorite character?
 - [character] -> Bot **will remember that**
- Do you know who [character] is?
 - No.
 - * [character] is a character [description] that appears in movies [movies he appears].
 - Yes.

ChatBot Concept

- * I love/hate that character.
 - * Do you like that character?
 - Yes/No -> **Bot will remember that**
- Have you seen [movie]?
 - No.
 - * [movie] was released in 1999 and is about [sinopsis].
 - Yes.
 - * Do you like that movie?
 - Yes/no -> **Bot will remember that**
- Tell me about [movie]
- Tell me about [character]
- Can you tell me more about [character]?
 - User inserts description -> Saved in DB
- In what movies has [character] appeared in?
 - User inserts movies -> Saved in DB

Design / Structure

A core aspect of the ChatBot is its mood, which will vary the type of questions and responses it does to the user, various activities may influence the mood. (Figure 4.1)



The main activity revolves around talking to the ChatBot about the Marvel Cinematic Universe, the bot starts with some knowledge of some Marvel movies and characters, and by speaking with the user it either complements or inserts new knowledge into a new Prolog database. In order to recognize text written by the user, it uses an extensive Prolog grammar developed by using the Wordnet Lexical Database, and to respond to the user it uses various response grammars. Which grammar is used depends on the bot's mood. User responses to the bot are analyzed using Senticnet and will affect the bot's mood. (Figure 4.2)

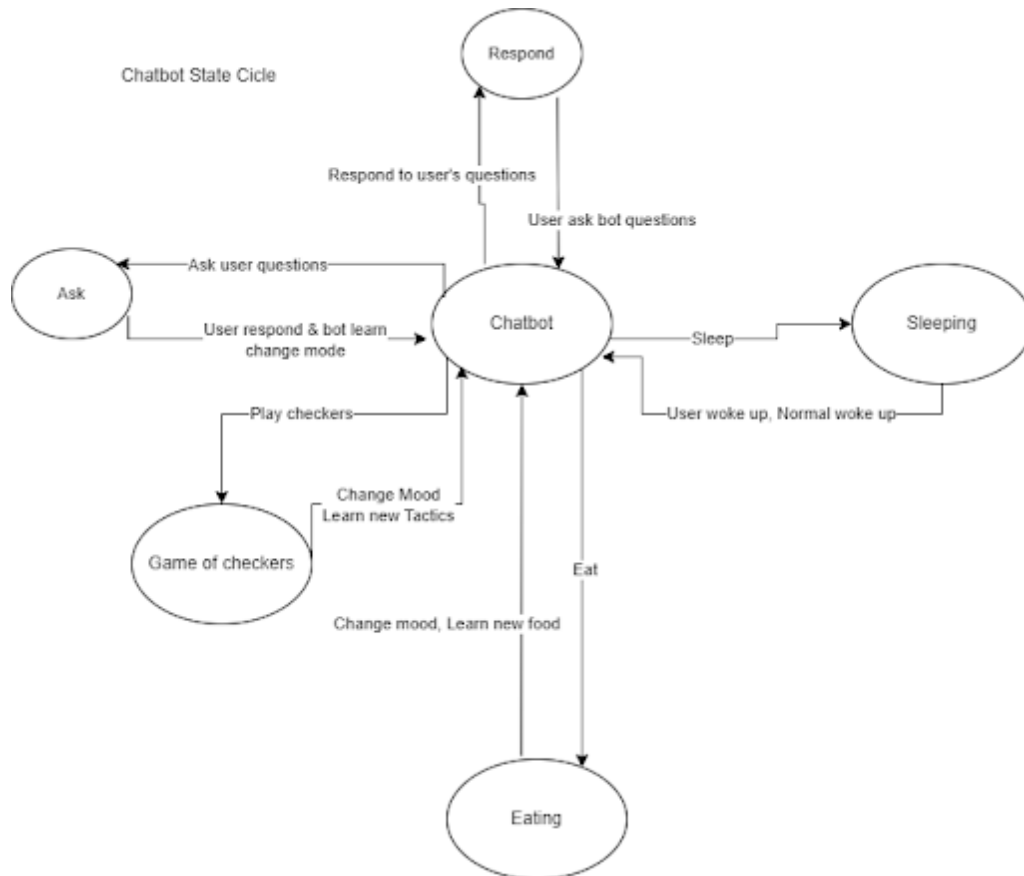


Figure 4.2: ChatBot States Diagram

Other activities include eating, sleeping, and playing checkers (this last one uses both Prolog and an OWL ontology).

Chapter 5

Implementation

5.1 Processing User Sentences

As previously mentioned, we used Spacy, to identify the key words of the answers of the user to properly understand the information and the Chatbot give a corresponding answer.

For this use with Spacy we arranged data to a format supported by Spacy and created a model. The process involves generating different sentences that include characters, movies and follow certain templates/syntax that the ChatBot supports, in other words, possible answers given by the user, configuring the *base_config* file provided in Spacy documentation, and then compile it, to finally be executed to start training the data, to obtain the model to be used.

To process the user sentence, we first check if it is a declarative sentence or interrogation sentence and we developed grammatics that are used in the main Python program with PySwip, for the possible sentences (declarative or interrogative) 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 as explained in the following Section.

In some of these topics the user can provide Information/knowledge, the method to accumulate the information will be discussed later in the Section.

5.2 Answering User

For the ChatBot to give different answers to the user, we used WordNet lexical database taking advantage of its amount of verbs, nouns, adverbs and adjectives.

Then for each conversation/interaction topic previously presented in Chapter 3 we developed grammatics that are used in the main Python program throw PySwip, for the possible answers that the ChatBot can give considering the context (Example: If the ChatBot didn't understand the sentence of the user "grammatically incorrect sentences, or sentences not supported by the system", we developed a grammatic, to produce an answer indicating that to the user 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. Regarding that, the ChatBot knowledge is saved in the file *botMemory.owl* using Owlready2.

When the ChatBot is initialized, it checks the file using the function *loadBotMemoryOnto()* and loads the different types of knowledge (user name, characters, movies, etc..) that contains knowledge of previous conversations with the ChatBot. After the 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 as we will explain in the following Section else doesn't understand the sentence and explains it to the user.

5.3 Ability to Accumulate Information/Knowledge

The Information/Knowledge of the ChatBot is saved in the file *botMemory.owl* using Owlready2. When the ChatBot is initialized, it checks the file using the function *loadBotMemoryOnto()* and loads the different types of knowledge (user name, characters, movies, etc..) that contains knowledge of previous conversations with the ChatBot.

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 also learn more information. In the end saves the new knowledge provided by the user and the results from the Web search in, (if any), in the *botMemory.owl* file.
- 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 into *botMemory.owl*.
- 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.

5.4 Tamagotchi Emotions

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.

Mood changes can either happen by use of using Senticnet on what it gets from the user, or by taking into account the last action by the user to the ChatBot.

Implementation

In the first case, the ChatBot 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.

In the second case, the ChatBot by interacting with the user leads to the creation of an event, represented in the form an enum, there are 3 actions that can happen, *likedFood*, *playAGame*, *dislikedFood*, this 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

5.4.1 Happy

The ChatBot will be happy as a combination of doing any of the various interactions available like being fed or by speaking with the user, it will get happier much faster if it's fed food it likes, or if it wins a game of checkers. With time, the ChatBot will grow less happy and become Content/Satisfied.

5.4.2 Satisfied

The ChatBot will be satisfied after being happy for some time, or after being sad and interacting with the user for some time, or if it was bored and was fed some of its favorite food. With time, the ChatBot will grow less content and become calm.

5.4.3 Calm

The ChatBot will be calm after being satisfied for some time, or after being angry and winning a game of checkers or eating food it likes, it can also become calm after waking up on its own from sleep, if angry it will also become calm after some time has passed.

5.4.4 Bored

The ChatBot will be bored after being calm for some time, it also will get bored if it's hungry or if it didn't like the food given, or if it lost a game of checkers, it will continue like this until it's either fed something he likes, in which it becomes satisfied if it goes to sleep and wakes up on its own, or if it interacts with the user by talking or by winning a game of checkers in which it becomes happy.

5.4.5 Angry

The ChatBot will be angry if it's bored and it's fed or wakes up early, it also gets angry if it interacts with the user while calm by losing a game of checkers or by eating something he doesn't like, if it does nothing after a while it will become calm.

5.4.6 Sad

The ChatBot will be sad if it loses a game of checkers while bored of it's satisfied and interacts with the user by talking, losing a game of checkers, or being given food it doesn't like, it will become satisfied if it wins a game of checkers, is given food he likes or talks about what he likes with the user.

5.5 Tamagotchi Interactions

5.5.1 Feeding

In the interaction Feeding it was developed a grammar for the ChatBot 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 method to save the knowledge of the foods and the feelings about them is the one explained in the previous section.

The way the ChatBot starves is by not being fed during an amount of time.

5.5.2 Sleeping

If the user doesn't interact with ChatBot during a certain amount of time, the ChatBot will go to a state of sleeping during an amount of time. If the user interacts with him while sleeping it can have the effects explained previously, such as it refusing to answer for an amount of time else if the ChatBot "awakes by himself" it restarts the counter of the time without interaction without side effects.

5.5.3 Playing

To play with ChatBot in this case Checkers we developed a grammar to support the commands that the user is capable of doing in the game. Finally, we 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 (example: Inform the color of Checker Piece or the state of the board).

As mentioned previously, playing the game and the end result will have an impact in the emotions of the ChatBot.

Chapter 6

Demo

6.1 Demo/Guide Main Conversation Theme

Note: Actions underline are the ones that will happen in the demo

- User: do you know who iron main is?
- Bot: Will say knows the character else will try to find it on the internet and will ask the user if he wants to know more about the character.
- User: tell me more
- Bot: Will say more about the character
- User: do you like him ?
- Bot: Knows the character and context of the question and will answer his opinion about the character.
- User: in which movie has iron man appeared in?
- Bot: Will say the movies that he knows the character appeared in ,else ask the user for help.
- User: iron man appeared in movies like iron man 1 and iron man 2
- Bot: Will thank the user depending on his mood and will learn the information provided by the user.
- User: in what movies that iron man appeared in ?
- Bot: Will say the movies that he knows the character appeared in ,else ask the user for help.

6.2 Demo/Guide Tamagotchi Interactions

- User: what do you hate to eat?
- Bot: Will say the foods he dislikes
- ...
- User: i want to play checkers
- Bot: Will initialize the game of checkers
- User: move a7 to 8b
- ...

References

- [1] Marvel cinematic universe. https://en.wikipedia.org/wiki/Marvel_Cinematic_Universe. Accessed: 03-05-2022.
- [2] Prolog. <https://pt.wikipedia.org/wiki/Prolog>. Accessed: 03-05-2022.
- [3] Senticnet. <https://senticonet.net/>. Accessed: 03-05-2022.
- [4] Senticnet api. <https://senticonet.net/api/>. Accessed: 03-05-2022.
- [5] Spacy 101. <https://spacy.io/usage/spacy-101>. Accessed: 2-06-2022.
- [6] Tamagotchi. <https://pt.wikipedia.org/wiki/Tamagotchi>. Accessed: 03-05-2022.
- [7] Use the actual eliza algorithm in your chatbot. <https://blog.csml.dev/use-the-actual-eliza-algorithm-in-your-chatbot/>. Accessed: 03-05-2022.
- [8] 90stoys. What is a tamagotchi, and why was it so popular? <https://www.90stoys.com/electronic-toys/what-is-a-tamagotchi/>, 4 2022. Accessed: 03-05-2022.
- [9] Gunther Cox. Chatterbot. <https://github.com/gunthercox/ChatterBot>. Accessed: 03-05-2022.
- [10] freeCodeCamp. Creating a chat bot. <https://www.freecodecamp.org/news/creating-a-chat-bot-42861e6a2acd>, 10 2014. Accessed: 03-05-2022.
- [11] Lamy Jean-Baptiste. Owlready2. <https://owlready2.readthedocs.io/en/v0.37/>. Accessed: 03-05-2022.
- [12] Amit Mathur. How to build eliza chatterbot a program that can chat with humans. <https://www.sourcecodesworld.com/articles/How-to-build-Eliza-Chatterbot.asp>, 12 2012. Accessed: 03-05-2022.
- [13] Python. Python. <https://www.python.org/>. Accessed: 03-05-2022.
- [14] Yüce Tekol. Pyswip. <https://pypi.org/project/pyswip/>. Accessed: 03-05-2022.
- [15] Princeton University. Wordnet a lexical database for english. <https://wordnet.princeton.edu/>. Accessed: 03-05-2022.