CentraleSupélec

Projet S7 – Moteur de dialogue et memóire à long terme

GONZALES VILLAROEL Tomas
MOUQED Mohamed Reda
NGUYEN HUYNH Titouan
REJAB Mohamed Rayen
STULBERG HUF Daniel

Supervisor: SEGUIER Renaud

20/01/2023

1. Introduction

Chatbots are becoming increasingly important in healthcare because they can help patients and healthcare providers in a variety of ways. Such growth accompanies the lack of trained medical personnel that became a concerning issue in these last years, as chatbots can be everywhere at once. One important area in which chatbots can have a significant impact is measuring patient's emotional states. Chatbots can determine whether patients are happy, sad, frustrated, or anxious by analyzing their language and tone. This information can be used to tailor the conversation and provide the patient with a more personalized experience. Furthermore, this data can be used to improve the overall patient experience by identifying and alerting clinical staff of patients who may be at risk of deterioration.

Chatbots can also offer emotional support to patients, which is especially useful for those who are isolated or suffering from mental health issues. They can also give patients self-care advice, assist them in managing their symptoms, and even help them navigate through the healthcare system. Chatbots can also help healthcare providers by answering patient's questions, providing relevant information, and relieving the burden on overburdened clinical staff. In short, chatbots can understand and respond to patient's emotional states, making them a valuable tool for healthcare providers looking to improve patient care and experience.

2. Ontology

Chatbots and ontologies are two related tools that have been increasingly used in a variety of industries. A formal representation of knowledge about a specific domain, such as healthcare or customer service, is an ontology. It is used to organize and structure data by defining the concepts and relationships within that domain. Chatbots, on the other hand, are computer programs that simulate human-to-human interaction. A chatbot can understand the context of a conversation and provide more accurate and relevant responses by using an ontology.

For instance, in the healthcare industry, a chatbot can utilize an ontology to comprehend the terminology that patients use to describe their conditions and offer correct information about possible treatments. Additionally, an ontology-based chatbot can be employed to recognize the user's mood and respond appropriately. Chatbots become more helpful and effective tools thanks to the use of ontologies, which also helps them comprehend and process information more effectively.

3. What is RASA

RASA is an open-source framework for developing conversational artificial intelligence. It enables developers to create custom chatbots and virtual assistants that can understand natural language inputs and respond contextually. RASA includes tools for natural language understanding (NLU), dialogue management, and response generation based on machine learning. It is compatible with a variety of messaging platforms and voice assistants, including Facebook Messenger, Slack, and Amazon Alexa. It aids in message comprehension, conversation, and connecting to messaging channels. It is also transparent, which means that we can see exactly what is going on underneath the hood and customize it as much as we want.

3.1. Intents

Intents in RASA refer to the underlying goal or purpose of a user's input. In other words, it is what a user wants to accomplish through his message, the action a user wants to take that they expect the chatbot to fulfill or facilitate. RASA NLU uses machine learning-based models to extract the intent from text. The intent recognition process is done by training the model with the set of labeled examples. Here are some examples below:

Hi, Hello, hey there => Intent: greet
Sad, unhappy, terrible => Intent: mood_unhappy
No, never, no way => Intent: deny

3.2. Entities

In RASA, entities are specific pieces of information that are extracted from the user's input. They can be used to provide more context for the intent and to customize the response. For example, an entity could be a location, a date, or a specific item. RASA NLU uses machine learning-based models to extract entities from text. We defined custom entities and train the model to recognize them.

```
entities:
  - name
  - mail
  - social_security_number
  - birth_date
  - gender
  - hobby
  - stress
  - time
  - location
```

```
- intent: give_mail
  examples: |
    - My email is [example@example.com](mail)
    - [random@example.com](mail)
    - Please send it to [anything@example.com](mail)
    - Email is [something@example.com](mail)
    - [paul312@gmail.com](mail) is my email
    - Sure it's [malikh.nj-mol@hello.fr](mail)
    - it is [brainiac@superman.uk](mail)
    - [example@email.com](mail) is my address
```

New entities defined

3.3. Actions

Actions in RASA refer to the steps taken by the rasa model, in response to a user's input. These actions are determined by RASA Core and executed by the RASA Action Server. Actions can include sending a message or performing other tasks.

```python
class ActionSetLatestHealthForm(Action):

    def name(self) -> Text:
        return "set_latest_health_form"

    def run(self, dispatcher: CollectingDispatcher,
            tracker: Tracker,
            domain: Dict[Text, Any]) -> List[Dict[Text, Any]]:

        userID = tracker.slots.get("social_security_number")
        Date = date.today().strftime("%d/%m/%Y")
        localisation = tracker.slots.get("location")
        SleepingTime = tracker.slots.get("start_sleep")
        WakeUpTime = tracker.slots.get("end_sleep")

        cursor.execute(f"INSERT INTO User_Dynam VALUES ('{userID}', '{Date}','{localisation}','{SleepingTime}','{WakeUpTime}')")
        conn.commit()
        return []
```

Custom action defined in Python

```
responses:
  utter_greet:
  - text: "Hey! Welcome to the Health chatbot! Is it your first time?"

  utter_happy:
  - text: "Great, carry on!"

  utter_goodbye:
  - text: "Bye"
```

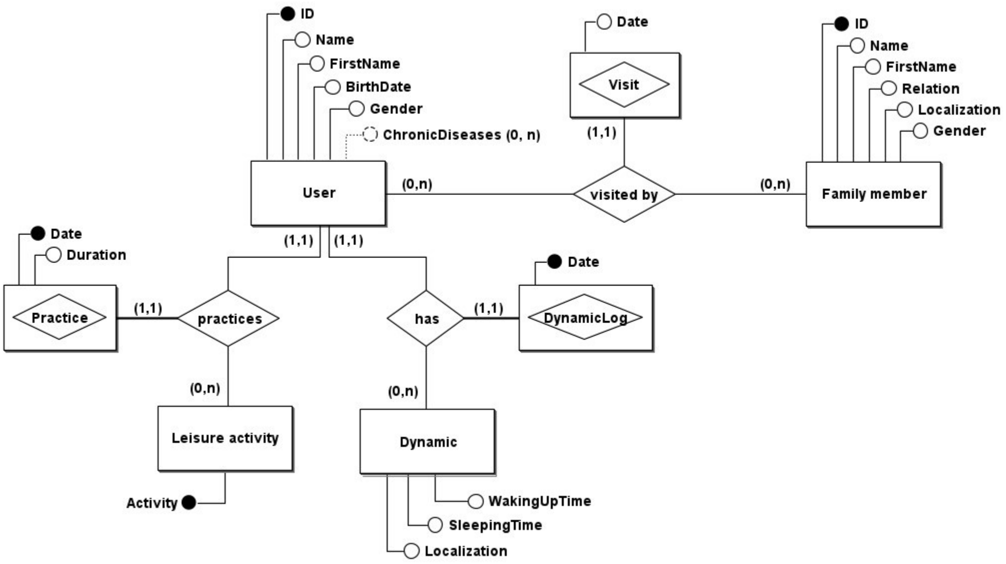
Example of utters, simple actions merely sending a message, that can be created using Rasa directly

3.4. Policies

The policies are the inherent rules that one chooses to impose on their rasa model. They can rule close to every aspect of its inner workings, but at a complexity cost. Indeed, these files act on a lower level than the outer rasa layer that is comparatively easy to operate and modify. Therefore, the integration of new inner features can take longer but it didn’t stop us from trying.

4. Database

Parallel to the process of developing the Rasa chatbot, part of this project was also focused on modelling, creating and populating our database. In order to better understand the problematic before starting to code the questions, a conceptual model of the database was first created. It can be seen in the figure below.



Conceptual model of our database

After the conceptual model, a database was created using the SQLite library. Such database is dedicated to store all the information that will be retrieved by the chatbot while talking to the patients. The database is structured as follows:

- User_Stat (UserId, Nom, Date de naissance, Sexe, Maladies chroniques)

  [table containing the static attributes of a given patient]

- User_Dynam (UserId, ChatDate, Localisation, SleepingTime, WakingUpTime)

  [table containing the dynamic attributes of a given patient]

- Famille (UserId, FamilyId, Nom, Relation, Localisation, Sexe)

  [table containing information about the relatives of the patients]

- Visites (UserId, FamilyId, Date)

  [table containing information about the visits of the relatives in given dates]

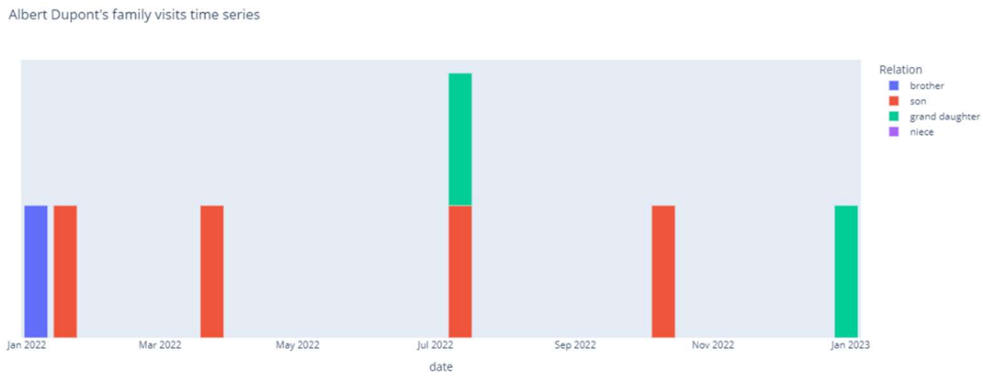
- Loisirs (UserId, Date, Activité, Opinion)

  [table containing information about leisure activities practiced by the patients]

We will enter in User_Stat the data typifying a specific user, in User_Dynam intel asked on a regular basis to a user in order to keep track of their current health. Data stored in Visites, Loisirs, and Famille will complete these and be filled through the different interactions with the user. The queries for these entries are particularly worth mentioning because their order would benefit treatment from the policy discussed with Emobot, influencing the probabilities and direction of the discussion. (Should we censor / redact this part?). An also important thing to denote is that the Loisirs table has actually been extended to all kinds of activity, and the opinions given about those are regularly updated, and can be used for sentiment analysis.

5. Visualization

After developing the database and populating it with a few lines of information, some visualization tools were created using a Python notebook. The purpose of making this type of resource available is to provide new insights that would not be noticeable by simply looking at the lines in the database tables. Therefore, such visualization tools are shown below.



It is valid to mention that such data was manually inserted into the database, and does not reflect any real answer from a real patient.

6. Limits and improvements

At the moment, a limiting factor in the effectiveness of our model rests in the detection of specific entities in the middle of a message. While we can work around this problem for most of the interesting information, tests on our first version have shown that name detection still poses a problem, as well as the detection of activities that the model has never encountered before. Yet non fully comprehensive, a long enough look up table (list of possible values that can be recognized as a certain entity) can be a solution for the unrecognized activities.

Name recognition performances can be increased by the importation of pretrained models as have shown our preemptive tries, even with foreign names. The use of such models however raises intellectual propriety questions, that should be settled with Emobot before further implementation.

Finally, the implementation of our custom policy to change the probability of some messages is finalized, but is slowly coming together, and should be able to fulfill the role we intended for it once completed.