**Week 5-6 – Adding entities, slots, responses and stories**

**Mathiopoulou Alexia A.M: 7115182100017**

**New Stories**

1. **Movie\_recommendation**

For the next part of my movie recommendation bot, I decided to continue by enriching the story that I had already created in the previous part. To do that, I had to add some slots and entities.

For the movie recommendation story, I firstly added the “utter\_movie\_recommendation\_path”, because as I stated in the end of the previous report, I intend to also add a sign\_up path that the user will be able to choose. With the utter\_movie\_recommendation\_path, the bot states to the user that he chose this dialogue path.

Next, I created the movie\_genre slot that would be filled by the movie\_genre entity. This entity would be filled by the intent movie\_genre, where the user would state his preferred movie genre after the bot’s respective question. I then added an action, with which the bot confirms the user’s preferred movie genre. In figure 1 below, we can see the final form of the movie\_recommendation story after those changes.

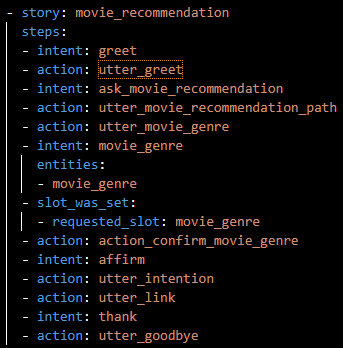


Figure 1. movie\_recommendation Story

For the bot to be able to confirm the user’s preferred movie genre, I had to add a simple custom action in the actions.py file. That’s why I created the “ActionConfirmMovieGenre” class, which retrieves the movie\_genre entity from the movie\_genre slot and returns it to the user. If the user doesn’t state his preferred genre, then the bot responds with the message “You did not tell me your preferred movie genre.”

Here we can see an example of a dialogue for movie recommendation:

Your input -> hi

Bot -> Hey! How can I help you?

Your input -> i want to watch a movie

Bot -> You chose the movie recommendation path.

Bot -> What movie genre do you like?

Your input -> drama

Bot -> Your preferred movie genre is drama.

Your input -> yes

Bot -> I will give you a link, where you can look for a movie to watch.

Bot -> https://www.imdb.com/list/ls031037940/

Your input -> thanks

Bot -> Have fun! Bye!

We can see that after the user states that he wants to see a movie, the bot states that he chose the movie recommendation path and asks him for his preferred movie genre. Then, the user answers with “drama” and the slot movie\_genre is filled with the entity value “drama”. That’s why the bot answers with “Your preferred movie genre is drama”. The story that follows is the same as the simple movie recommendation story that I presented in the previous report; the bot informs the user that it will give him a link from a movie website and then proceeds to give the link.

1. **Create\_user**

As I couldn’t develop the movie\_recommendation path any more without adding more complicated custom actions (I will do that in the next part), I decided to create a **create\_user** story path, to explore some other of Rasa’s capabilities. With this story, I wanted the bot to be able to ask some for some personal information from the user, such as his name and e-mail. To do that, I knew that I had to add, again, some slots and entities. My initial thought was to design a simple story, where the user and the bot would greet each other, the user would state his desire to create an account or sign up and the bot would ask for his information. To do that, I created the *request\_sign\_up* intent (figure 2).

Text

Description automatically generated

figure 2. «request\_sign\_up» Intent

For the bot’s responses, my first thoughts were to just add a simple action where the bot would ask from the user some personal information, such as his name and e-mail address. However, as I previously mentioned, I thought that it would be nice and would make the information more interesting if the bot would recognize different conversation paths , such as a sign up path and a movie\_recommendation path. After some research, I found out that I could do that by creating some forms and that’s why I created the *sign\_up\_form* in the domain.yml.

The sign\_up\_form would require two basic slots to be complete, *name* and *email* (figure 3).

Figure 3. sign\_up\_form

**Text

Description automatically generated**

**Text

Description automatically generatedText

Description automatically generated**These two slots would have the form of a text, they wouldn’t influence the conversation and the mapping would be from\_text, contrary to the movie\_genre slot that was filled from\_entity. Below (figures 4 and 5), I present the slots name and email, that I created by following this logic:

Figure 5. “email” Slot

Figure 4. “name” Slot

We see in the above figures that in both cases there is an active\_loop option, which is the sign\_up\_form. This means that, in order for the slots to be filled, the sign\_up\_form must be active. To activate the sign\_up\_form, I had to set up a rule in the rules.yml with the name “*Activate sign\_up form*”, which states that after the “utter\_signup” action, the sign\_up\_form is triggered and activated.

The bot’s responses for this story are the actions utter\_signup, and utter\_aknwoledge\_info.

The story is developed as we can see in the following figure:

Text

Description automatically generated

Figure 5. "create\_user" Story

We notice that after the action “sign\_up\_form” the form is activated with the active\_loop option and then the required slots name and email are applied. After that, the bot acknowledges the user’s info and says goodbye.

**A simple example dialogue for the sign\_up form:**

Your input -> hi

Bot -> Hey! How can I help you?

Your input -> i want to create an account

Bot -> You chose to sign up.

Bot -> Could you please provide your name?

Your input -> alexia

Bot -> Could you please provide your email?

Your input -> alexm@gmail.com

Bot -> Thanks alexia!

Bot -> Have fun! Bye!

**Comments:**

As we see, the only recommendation that the bot does is the links that it gives to the users. For the next stage of the bot’s development, I will try to make write actions in actions.py file, to take movie data form a website, such as IMDB, in order to make the bot able to suggest real movie titles to the users.