**Week 7-8 – Adding actions**

**Mathiopoulou Alexia A.M: 7115182100017**

**Actions**

The next stage for the bot’s development is the addition of custom actions. In the previous part I have already added a simple custom action that retrieves the movie\_genre entity from the movie\_genre slot and returns it to the user. In this part I want to create more complicated action that takes information from a website in order to make actual movie recommendations to the user. To do that, I decided to use the IMDB website, because it has a large data base of movies, tv shows and video game.

When I started writing my action, I thought that it would be better if I used the imdpy library that contains all the movie information that I needed for my bot. However, after its creation I encountered some problems. More specifically, while the story run flawlessly, without any corrections from my side, in rasa interactive, when I ran it in rasa shell it returned me a ClientPayloadError. After some more trial I decided to drop the idea of using the imdbpy library.

I then proceeded in using the web scraping method to take information from the IMDB website in order to recommend movies to the user. To do that, I used the BeautifulSoup feauture to parse and extract the needed information from the IMDB website (by using the url of the asked movie genre each time). I then made a list of the movie titles for each genre; I decided to choose only English movie titles because the first time I ran the code without this clarification, some movie titles were in greek, something that later produced errors (I also wanted everything to be in english). From the list with the movie titles, the bot chooses 5 random movies to recommend to the user each time (for the asked genre).

**Movie Recommendation**

After creating the “actions\_make\_movie\_recommendation” action, I added it in the domain.yml file and the next step was the enrichment of the movie recommendation stories. I created a new story “actual\_movie\_recommendation” that was the same as the movie\_recommendation story presented in the previous report, but I was about to change that. This time, after the “action\_confirm\_movie\_genre”, I remove the actions that gave a link to the user and I added the new custom action that I created. Then, the bot checks if the user likes the recommendation with the “utter\_check” action. If the user affirms or thanks the bot (I used an **or** statement here) then the bot says goodbye.

Below we can see the story (figure1) and a dialogue example (figure 2):

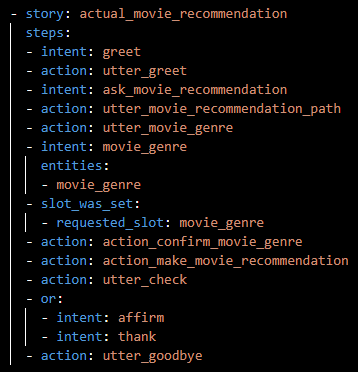
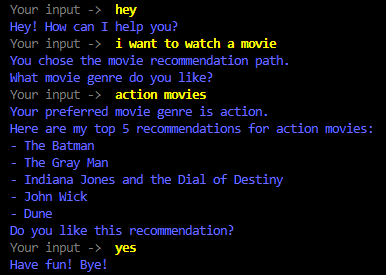


Figure 2. "actual\_movie\_recommendation" Dialogue

Figure 1. "actual\_movie\_recommendation" Story

As we see in figure 2, the user states that he wants to watch a movie, the bot informs him about the movie recommendation path and asks him for his preferred movie genre. The user then answers with action movies and the bot correctly responds that his preferred movie genre is action. Next, we notice that the bot recommends 5 action movies. Finally, it asks the user if he likes the recommendation and after his affirmation it says goodbye.

I also run some dialogues in rasa interactive (all interactive stories are commented at the bottom of the stories.yml file) to see if there are any problems with the entities or anything else. I tried to change some things, to see how the bot would respond. For, example after the “utter\_check” action, I tried the intent “deny” instead of the “affirm” one. That’s when I came up with the idea (while running interactive\_story\_4) to add a second movie recommendation (figure 3). More specifically, if the user answers with the intent “deny” after the “utter\_check” action, then the bot repeats the “action\_make\_movie\_recommendation” action, checks again and then the dialogues continues as before. However, I also wanted to give the opportunity to the user to continue or restart the conversation if he wants. That’s why I added the “utter\_continue” action, with which the bot ask the user if he has any other question. If the user responds negatively, then the bot uses the “utter\_goodbye” action, but if the user responds positively, then the bot uses the “action\_restart” action and the dialogue restarts.

To use the “action\_restart” action, that is a default Rasa action, I had to add the ActionRestart action in the actions.py file.

Below we see the four new stories that I created based on the above description:

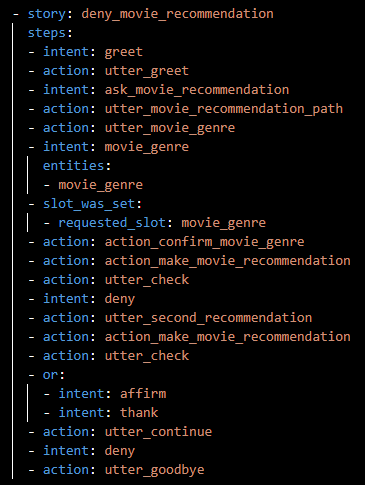
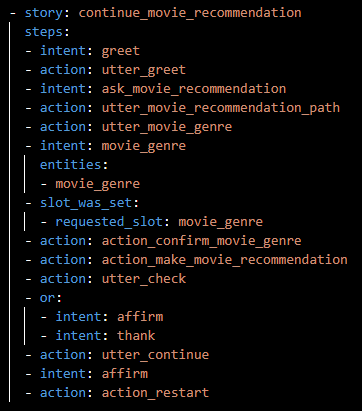


Figure 4. "continue\_movie\_recommendation" Story

Figure 3. "deny\_movie\_recommendation" Story

In figure 3, we have the case where the user doesn’t like the first movie recommendation that the bot gave him and then the bot makes a second one. The bot then asks for more questions and the user denies it. The dialogue ends. In figure 4, we have one movie recommendation, but the user wants more questions, and the dialogue restarts.

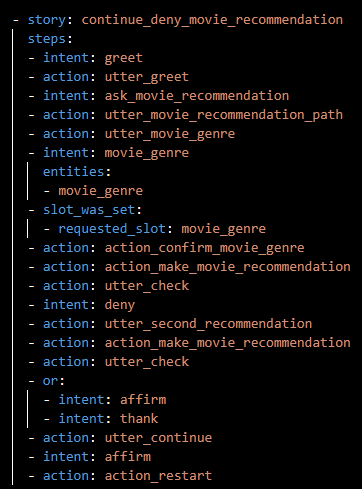
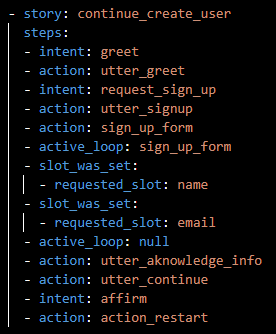


Figure 6. "continue\_create\_user" Story

Figure 5. "continue\_deny\_movie\_recommendation" Story

In figure 5, we have both a second movie recommendation and restart of the dialogue. Finally, in figure 6 we see the story that I created for the create\_acount story path. In this case the user can also continue the dialogue and ask for a movie recommendation after the end of his first dialogue with the bot.

We can now see some dialogue examples for the stories presented above:

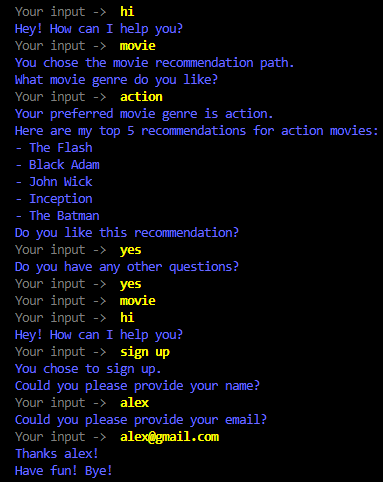
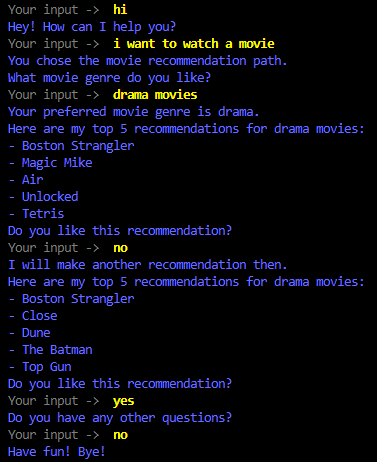


Figure 8. "continue\_movie\_recommendation" Dialogue

Figure 7. "deny\_movie\_recommendation" Dialogue

Finally, I present below the pipeline and the policies that I used in the config.yml file, for the designing of my bot. I have commented the use of each feature used in the config.yml file.

Text

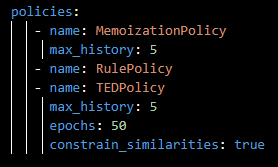
Description automatically generated

Figure 9. pipeline

Figure 0. policies

**Conclusions:**

After many trials with different dialogues and all the movie genres that I could think of, I can say that the recommendations that the bot does are pretty accurate and its overall performance is actually good.

Some ways that the bot could be improved, is maybe with the addition of some buttons for entities and slots. For example, all the movie genres could be buttons that the user could choose from. Furthermore, the bot would be more interesting if the user could ask for as many recommendations as he would like and as many times as he wanted. Finally, it would be nice if this exact bot could be used for tv series or even music recommendations with some alterations in the “ActionMakeRecommendation” custom action. But these are just thoughts for future work.