

TARS Chatbot Report

By: Sameer Haider and Priyesh Patel

Introduction

Our chatbot is known as TARS. The name is inspired from the actual artificial robot that was portrayed in *Interstellar*. TARS is able to answer general questions about the movie *Interstellar*. It can answer specific questions within the knowledge domain around the movie, as well as some questions about the cast. Possible questions consist of “what is interstellar”, “who’s the director”, “who’s in the cast” etc... We were able to make the chatbot flexible, allowing multiple variations of input to the same intent. This allows more of a dynamic feel.

As for the development of our chatout we decided to use AIML, an XML-based language that was developed by the A.L.I.C.E AI Foundation and Richard Wallace. AIML provides a simple way to develop responses to user info in order to build a chatbot interface. We used the PyAIML library in python to interact with the AIML file. Through our chatbot creation we started to see more and more issues with AIML’s pattern matching. The pattern matching was so strict that any slight spelling mistakes would cause the chatbot to not understand the user's request. We were able to solve these issues using some NLP techniques that we learned from class.

Instructions for setting up Chatbot

1. First clone the repo:

git clone <https://github.com/sameer-haider/HLT-chatbot-project.git>

2. Open inside an IDE, then create a virtual environment with python version 3.7 using the following command: (install python-3.7 first if you don't have it)

python3.7 -m venv <venv-name>

3. Source it with the following command. You can also do it through your IDE

Mac: **source <venv-name>/bin/activate** or Windows: **<venv-name>\Scripts\activate.bat**

4. Install all the required packages & respective versions with the following:

pip install -r requirements.txt

5. Run main.py

NOTES if an error related to `en_core_web_md` occurs, remove the first line (starts with "https") in `requirements.txt` and run this command in the terminal. Then do step 4 again:

python -m spacy download en_core_web_md

System Description

NLP Techniques

We used multiple NLP techniques to improve the functionality of our chatbot.

Libraries used: nltk, spaCy, and numpy.

1. Similarity Matching

We used a combination of nlp techniques in order to match user input to the patterns.

- a. Remove extraneous characters

Remove punctuation and other characters, only keeping alphanumeric characters.

b. Tokenization

Tokenize the user input as well as all the intents in the base.

c. Lemmatization

Lemmatize each token, turning it into its base form

d. Calculate Tf-vectors

Calculate the term frequencies for the user input and each pattern/intent

e. Calculate Cosine Similarity scores

Calculate the cosine similarity between the user_input and each pattern. We store the highest one.

f. Thresholding

Return the highest-score pattern if the cosine similarity score is above the threshold. We picked the threshold value after testing and tweaking our chatbot.

2. Named Entity Recognition

We used the powerful spaCy library to implement Named Entity Recognition

a. Download database

Load the *en_core_web_md* database from spaCy

b. Process message

Process the message with the spaCy module, which does the main NER computation

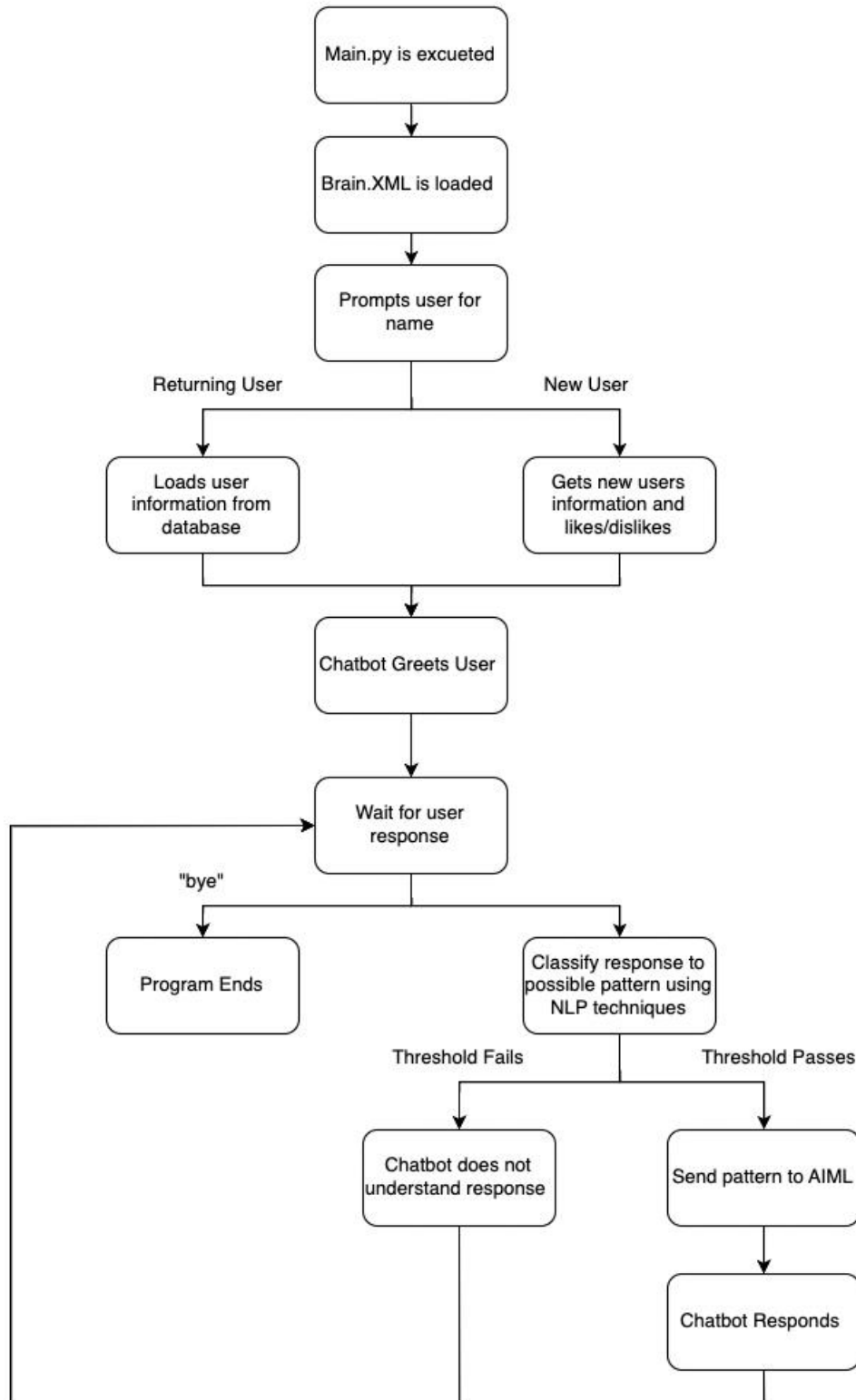
c. Retrieve named entities

Retrieve the named entities + type and return. Return None if no named entities.

After retrieving the named entities, we use them to customize certain responses like “*What character did [actor] play?*”

Depending on the actor (which is identified with NER), we return a specific response.

Diagram of Dialog & Logic



Sample Dialog Interactions

1. Existing User

```
Enter a username, or enter "new" if you want to make a new user: Tom
TARS: Hello! It's nice to see you again Tom!
USER: Hi
TARS: Hello!
USER: How's it going
TARS: Im functioning!
```

2. New User

```
Enter a username, or enter "new" if you want to make a new user: new
Choose a username (chatbot will address you by this): John
Do you like space science. Interstellar talks a lot about space. Enter (y/n):y
Do you like Matthew McConaughey? Enter (y/n):n
TARS: Nice to meet you, John! My Name is TARS would you like to talk about Interstellar?
USER: Hi
TARS: Hi there John!
```

3. Full Conversation involving NER, Interest based reponses, and ending conversation

```
Enter a username, or enter "new" if you want to make a new user: Jim
TARS: Hello! It's nice to see you again Jim!
USER: What can you tell me?
TARS: I can provide you with some facts about Interstellar.
USER: Did Matthew McConaughey win any awards from this movie?
TARS: I know that you like Matthew McConaughey, so you should be proud to learn that he won the Academy Award for Best Actor, as well as other awards!
USER: Cool!
TARS: Very Chill!
USER: How long is the movie?
TARS: The movie is quite long compared to the average runtime of a movie it has a runtime of 2 hours and 49 minutes.
USER: Could you tell me what Interstellar is about?
TARS: I can provide you with some facts about Interstellar.
USER: What is Interstellar about?
TARS: The movie is about a group of astronauts who travel through a wormhole in search of a new home for humanity after Earth becomes uninhabitable. Along the way, they discover the secrets of black holes and the power of love.
USER: What are black holes?
TARS: I know you hate space science, so here is the simple explanation for this super boring topic. Black holes are, like, these scary things in space that pull everything in.
USER: Who did Matt Damon play in Interstellar?
TARS: Matt Damon played Dr. Mann in Interstellar.
USER: Who did Michael Caine play in the movie?
TARS: I'm sorry, I don't know who MICHAEL CAINE played in Interstellar.
USER: Thank you for the information!
TARS: You're welcome!
USER: Bye!
TARS: Goodbye! Have a nice day Jim.
```

Appendix A - Knowledge Base

Since we decided to work with AIML we needed a brain or knowledge base XML file. This file is used to match patterns and return appropriate responses. We were able to use information gained from the web crawler assignment to create responses to the user's questions. One drawback to our knowledge bases is that it is fairly small mainly due to our topic.

Below is snippets from the Brain.XML file

```
<category>
  <pattern>WHAT ARE BLACK HOLES</pattern>
  <template>Black holes are created when a star dies and collapses in on itself.
    Right at the moment of its creation, a gigantic explosion is released which is called a supernova.
    This supernova creates so much light and energy that it can be seen billions of light years away!
    After a black hole is born, it will suck in all matter and light through gravity and grow as
    it consumes more and more.</template>
</category>

<category>
  <pattern>WHAT GENRE CATEGORY DOES THE MOVIE BELONG TOO</pattern>
  <template>Interstellar is a science fiction movie.</template>
</category>

<category>
  <pattern>WHICH WHO ACTORS PLAY ACT IN THE MOVIE</pattern>
  <template>. The movie features a cast of actors, including Matthew McConaughey,
    Anne Hathaway, Jessica Chastain, and Michael Caine, among others. </template>
</category>

<category>
  <pattern>HOW LONG IS THE MOVIE</pattern>
  <template>The movie is quite long compared to the average runtime of a movie it has a
    runtime of 2 hours and 49 minutes.</template>
</category>
```









```
<category>
  <pattern>HELLO</pattern>
  <template>
    <random>
      <li>Hi there <get name="name"/>!</li>
      <li>Nice to meet you <get name="name"/>!</li>
      <li>Hello, how can I help you?</li>
      <li>Hello!</li>
      <li>What's up!</li>
    </random>
  </template>
</category>

<category>
  <pattern>NEW USER *</pattern>
  <template>Nice to meet you, <set name="name"><star/></set>! My Name is TARS would you like to talk about Interstellar? </template>
</category>

<category>
  <pattern>RETURNING USER *</pattern>
  <template> Hello! It's nice to see you again <set name="name"><star/></set>! </template>
</category>
```

Appendix B - User Models

We store the user information in the user_models SQL table. The table is stored locally in the user_models.db file, with test users already inserted. The table looks like the following:

user_models		CREATE TABLE user_models (id INTEGER
 id	INTEGER	"id" INTEGER
 username	TEXT	"username" TEXT NOT NULL
 age	INTEGER	"age" INTEGER NOT NULL
 like_1	INTEGER	"like_1" INTEGER DEFAULT 0
 like_2	INTEGER	"like_2" INTEGER DEFAULT 0
 like_3	INTEGER	"like_3" INTEGER DEFAULT 0
 like_4	INTEGER	"like_4" INTEGER DEFAULT 0
 like_5	INTEGER	"like_5" INTEGER DEFAULT 0

All of the like_n store binary 0 or 1, to show if the user likes or dislikes a certain topic as determined in the code. We use the SQLite3 library in python to add new users to the table and also retrieve information to use in the chatbot.

Evaluation and Analysis

TARS is a simple chatbot we created that can provide a user with limited information and facts relating to the movie Interstellar. We plan on refining dialog and building a complex chatbot that can pick up on a user interest through dialog alone. Currently, TARS can provide limited information about Interstellar such as plot, cast, and awards. Due to the limited amount of information one movie can provide we would like to expand the chatbot to work with all movies instead of just one in the future. This would provide us with a larger knowledge base. A limited knowledge base was very hard to work with so a larger one would help.

We were able to create the TARs chatbot almost completely from scratch with the help from AIML. TARs has some strengths which include a clean user interface, named entity recognition to identify actor names, and a cosine similarity score to classify the users response. In addition, based on the user's likes or dislikes TARs can display specific responses that would align with those likes or dislikes. As for weaknesses, TARs has a very limited knowledge bases

so the responses are also very limited. Another weakness we faced was using AIML. Initially, using the AIML pattern match was very useful. However, as our responses got more and more complex, AIML was very strict and couldn't determine the appropriate response. Our solution was to use NLP techniques such as cosine similarity, TF-vectors, and threshold scores to identify the closest related response.

As for future improvements, we would like to expand our current chatbot to have conversations about movies in general instead of only Interstellar. For this we would need a larger knowledge base. We would try to implement a web live look-up compared to a static knowledge base. With the issues faced using AIML we would also try a different form of pattern matching and response. Another improvement we want to make is to update the user models with more information and have the chatbot respond uniquely based on the users current mood. Overall, there are a lot of optimization and functionalities we could incorporate. We could also incorporate machine learning to the chatbot so that it can learn based on each user.