HUMAN MACHINE DIALOGUES PROJECT REPORT

Manh Tuan NGUYEN

Master 2 Student in AI
Trento University – Avignon University

Table of Contents

Acknowledgement

I would like to express my gratitude to all those who gave me the possibility to complete this thesis in specific as well as academic support in general that I received during the time of my studying as an exchange student at the University of Trento.
I wish to express my sincere thanks to my professor Giuseppe Ricardi for providing me and other students with all the necessary facilities for the research.
Foremost, I would like to send my honest thanks again to professor Ricardi , Mr. Giuliano Tortoreto and Mr. Seyed Mahed Mousavi for sharing expertise , and sincere and valuable guidance and encouragement extended to me during the time of the course.

I. Introduction

Conversation has been and will always be one of the most important parts in human life. With the progress of mankind, people have applied technology to communicate with each other through electronic devices such as laptops, smartphones,… Now, we have even more than that thanks to Artificial Intelligence. Humans began to make simple communication with the “chatbot” or “virtual assistant”. By combining Artificial Intelligence with Natural Language ideas, we already had such a powerful assistant like Alexa, Google,… which can easily answer your question or complete some simple task from users. This course – Human Machine Dialogue (HMD) is aimed at providing knowledge and let’s students make a simple virtual assistant on their own. This report will describe my virtual assistant as a final project of HMD.

1. **Model type:**
   I chose to develop a question-answering model type. User will ask a question relative to the domain and my model will give the answer back.
2. **Domain:**
   My model has 3 main domain that can be answer :
   - **Movie information:**
     - ***Input:*** Question about movie information such as: Who is the director of Bad Boy?
     - ***Output:*** The director's name in a full sentence.
     - ***Describe:*** This will be the core function of my model since my model is specialized in this domain. The model covers all the information of the movie from the crew, year , actor,… to the recommendation system by the desired genre of the user.
   - **Chit chat :**
     - ***Input:*** Simple opening question about the age , health, function of the model and some controversial question about gods, lgbt community and love or also tell some joke or ask about the time. Example: I love you, Do you love me?
     - ***Output:*** A funny and predefined answer will be given to the user. For example: I'm truly in love with you despite the machine part or If I was a human, definitely I would chase after you.
     - ***Describe:*** This function will be the second core of the model. A simple chit chat option will bring the model more alive due to my experiment with Google Assistant. Because sometime during the conversation, users can get bored and ask some chit-chat non relative questions to the current talk so this function is to prepare for that situation.
   - **Weather information:**
     - ***Input:*** Users ask questions about the general weather in a specific city (default will be Trento). Example: What is the weather today? Or how’s the weather today?
     - ***Output:*** The assistant you answer back with corresponding information about the weather which include both temperature, real feel-like temperature and the condition. For

example: The current temperature in Trento is 9 degrees but it actually feels like 6. The weather is cloudy.

3. **User group:**
   Due to all the functions defined above, the target users of this system will be the users who want to find information about the cinema industry but also sometimes want to play with the model just for fun.

II. Conversation Design

To begin building the model, I have to think of a general idea about conversation design. To think about that, I have to prepare the idea of what topic domain, what kind of dialogue my assistant can take as input. And from that, what property should I have in order for my assistant to make the answer.

1. **Dialogues:**
   The dialogue should be simple. User gives a command/ talking and the assistant takes turns to answer. Each phrase from the user should be indicated directly to the topic covered of the model.
2. **Properties:**
   All the properties should be defined in the third part. The most important thing is that I need to also indicate the Fallback property so that if the users make any non-relevant information, the assistant should know how to respond. This assistant will be a mixed initiative type.

III. Data Description and Analysis

1. **Dialogue data:**
   i. **Creation:**
   The dialogue data is created by myself based on the testing idea when I tried to make conversation with Google Assistant and with my friend in real life. The idea when creating these dialogues is to try to cover all the possibilities of questions relevant to the domain of the model.

   ii. **Analysis:**
   The way I created these data is strictly following the conversation data instruction of rasa with rules, stories slots and entities.
   - *Rules:* 6 rules. All the rules are created to reduce the time needed to create the storyline.
     - Say goodbye when users say goodbye
     - Answer chitchat whenever users chit chat
     - Say hello whenever users say hello
     - Ask information about genre whenever users ask for a movie recommendation.
     - Answer the time whenever users say times.
   - *Conservation flow – story:*
     - 26 stories in total cover all the happy path and sad path of each function
     - The longest story has 10 turns. The shortest one has 4 turns.
   - *Entities:*
     - 4 entities type in total: PERSON, genre, movie_title and city.

        - PERSON: Name of a person – could be an actor or director.
        - Genre: the genre of the movie users want to see.
        - Movie_title: the title of the movie which users want to ask for information about.
        - City: the name of a city where users want to know about the weather.
    - *Slots:*
        - 5 slots in total : PERSON, genre, movie_title , city and act. All of them will have an important impact on the conversation.
        - PERSON: get the relative entities information and store to be processed further.
        - Genre: get the relative entities information and store to be processed further.
        - Movie_title: get the relative entities information and store to be processed further.
        - City: get the relative entities information and store to be processed further
        - Act: The customize slot where the action will analyze the specific input of users and set the correct value for this slot. This slot is used for the movie information task.

2. **NTT data:**
    i. **Creation:**
    This NLU dataset is created and collected by myself. Most entities relevant to the movie function is collected from github. Others were created by myself based on the answers of Google Assistant and my friends' answers when I talked with them.
    ii. **Analysis:**
        - There are 23 in total.
        - The top intent has 80 examples and the lowest one only has 4 examples. On average , each intent has 14,6 examples.
3. **Domain data:**
The dataset about movies and about weather is completely connected through API. I will describe more about each API below.
    i. **Imdb API:**
    Imdb API is provided free on RapidAPI website (for 500 requests per month) . This API provides a lot of GET functions from information about actors to information about movies.
    ii. **Open Weather Map API:**
    This API is also provided free on RapidAPI website. This Api contains 6 GET functions to take the forecast information. I only use one to get the information about the current weather data at a specific location.

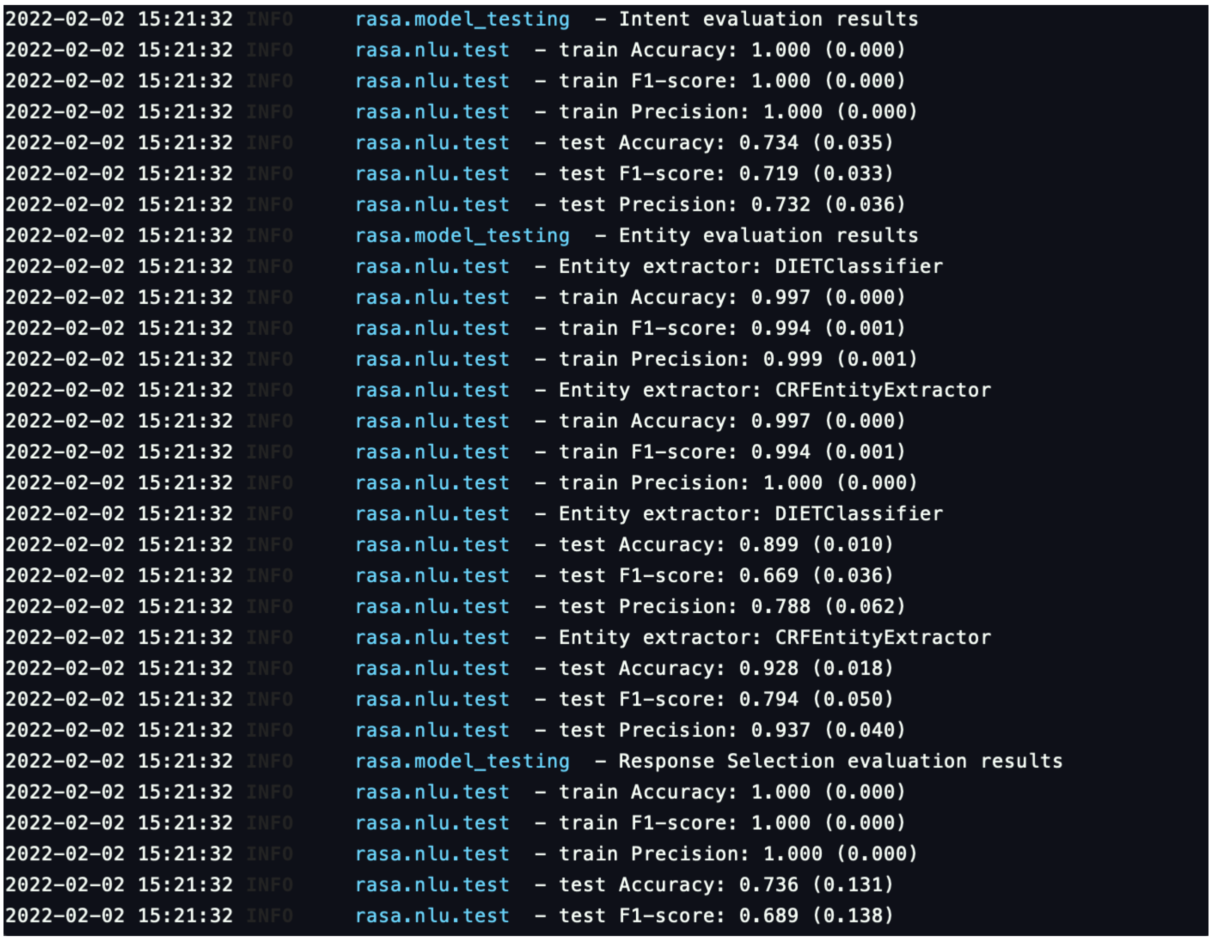
IV. Conversation Model

1. **Algorithm:**
This model is completely a rasa-based model system. I used the Rasa framework to create the model.

2. **The response policy:**
   In this project , I’m using 3 policies to decide actions to answer the user.
   - Memorization policy: This is a recommendation policy from Rasa. This policy helps to memorize the stories from training data which is really helpful when my story data is created by thinking of covering all the possibilities.
   - TED policy: The Transformer Embedding Dialogue, the top 1 of recommended policy from Rasa.
   - Rule policy: I added this policy to force the model to follow a fixed behavior which has been declared inside the rules file.
3. **The fallback policy:**
   My model uses the DIET Classifier and FallBack Classifier which already implemented the fallback policy. So whenever there is no classifier that can recognize the entity with high confidence, the DIET classifier will flag them as nlu_fallback entity. To overcome this problem, I also set a rule that if we face this situation, the machine will always answer the same and ask for changing the topic of the current conversation.
4. **The vocal model:**
   As I received recommendations from the HMD course’s team , I decided to use Alexa Skills as my vocal model. By connecting my rasa chatbot with Alexa, I have the ability to use all of the features of the Alexa vocal model. Also, the Alexa Skills also provide an interface that I can bring with me to ask people to test my system.
5. **The Natural Language Understanding model:**
   For NLU detection, I used 3 different Classifier:
   - DIET Classifier
   - FallBack Classifier
   - Spacy NLP

V. Evaluation

1. **Natural Language Understanding:**
   For this part, it’s really hard to ask users to evaluate this since it will be too complicated due to the technical term. So instead of using human resources, I’m using the command line provided by Rasa to perform the cross-validation to get the performance of the model with the number of splits being 5.



```
2022-02-02 15:21:32 INFO     rasa.model_testing  - Intent evaluation results
2022-02-02 15:21:32 INFO     rasa.nlu.test  - train Accuracy: 1.000 (0.000)
2022-02-02 15:21:32 INFO     rasa.nlu.test  - train F1-score: 1.000 (0.000)
2022-02-02 15:21:32 INFO     rasa.nlu.test  - train Precision: 1.000 (0.000)
2022-02-02 15:21:32 INFO     rasa.nlu.test  - test Accuracy: 0.734 (0.035)
2022-02-02 15:21:32 INFO     rasa.nlu.test  - test F1-score: 0.719 (0.033)
2022-02-02 15:21:32 INFO     rasa.nlu.test  - test Precision: 0.732 (0.036)
2022-02-02 15:21:32 INFO     rasa.model_testing  - Entity evaluation results
2022-02-02 15:21:32 INFO     rasa.nlu.test  - Entity extractor: DIETClassifier
2022-02-02 15:21:32 INFO     rasa.nlu.test  - train Accuracy: 0.997 (0.000)
2022-02-02 15:21:32 INFO     rasa.nlu.test  - train F1-score: 0.994 (0.001)
2022-02-02 15:21:32 INFO     rasa.nlu.test  - train Precision: 0.999 (0.001)
2022-02-02 15:21:32 INFO     rasa.nlu.test  - Entity extractor: CRFEntityExtractor
2022-02-02 15:21:32 INFO     rasa.nlu.test  - train Accuracy: 0.997 (0.000)
2022-02-02 15:21:32 INFO     rasa.nlu.test  - train F1-score: 0.994 (0.001)
2022-02-02 15:21:32 INFO     rasa.nlu.test  - train Precision: 1.000 (0.000)
2022-02-02 15:21:32 INFO     rasa.nlu.test  - Entity extractor: DIETClassifier
2022-02-02 15:21:32 INFO     rasa.nlu.test  - test Accuracy: 0.899 (0.010)
2022-02-02 15:21:32 INFO     rasa.nlu.test  - test F1-score: 0.669 (0.036)
2022-02-02 15:21:32 INFO     rasa.nlu.test  - test Precision: 0.788 (0.062)
2022-02-02 15:21:32 INFO     rasa.nlu.test  - Entity extractor: CRFEntityExtractor
2022-02-02 15:21:32 INFO     rasa.nlu.test  - test Accuracy: 0.928 (0.018)
2022-02-02 15:21:32 INFO     rasa.nlu.test  - test F1-score: 0.794 (0.050)
2022-02-02 15:21:32 INFO     rasa.nlu.test  - test Precision: 0.937 (0.040)
2022-02-02 15:21:32 INFO     rasa.model_testing  - Response Selection evaluation results
2022-02-02 15:21:32 INFO     rasa.nlu.test  - train Accuracy: 1.000 (0.000)
2022-02-02 15:21:32 INFO     rasa.nlu.test  - train F1-score: 1.000 (0.000)
2022-02-02 15:21:32 INFO     rasa.nlu.test  - train Precision: 1.000 (0.000)
2022-02-02 15:21:32 INFO     rasa.nlu.test  - test Accuracy: 0.736 (0.131)
2022-02-02 15:21:32 INFO     rasa.nlu.test  - test F1-score: 0.689 (0.138)
```

Figure 1: Cross-Validation

All the results are shown in the pictures. I used various classifier for NLU in order to guarantee that we can have the best correct result with high confidence. To be better visualize, below is the confusion matrix of CRF Entity Extractor, DIET Classifier, intent confusion matrix and response selection confusion matrix. For better visualize , please refer to my github link here.

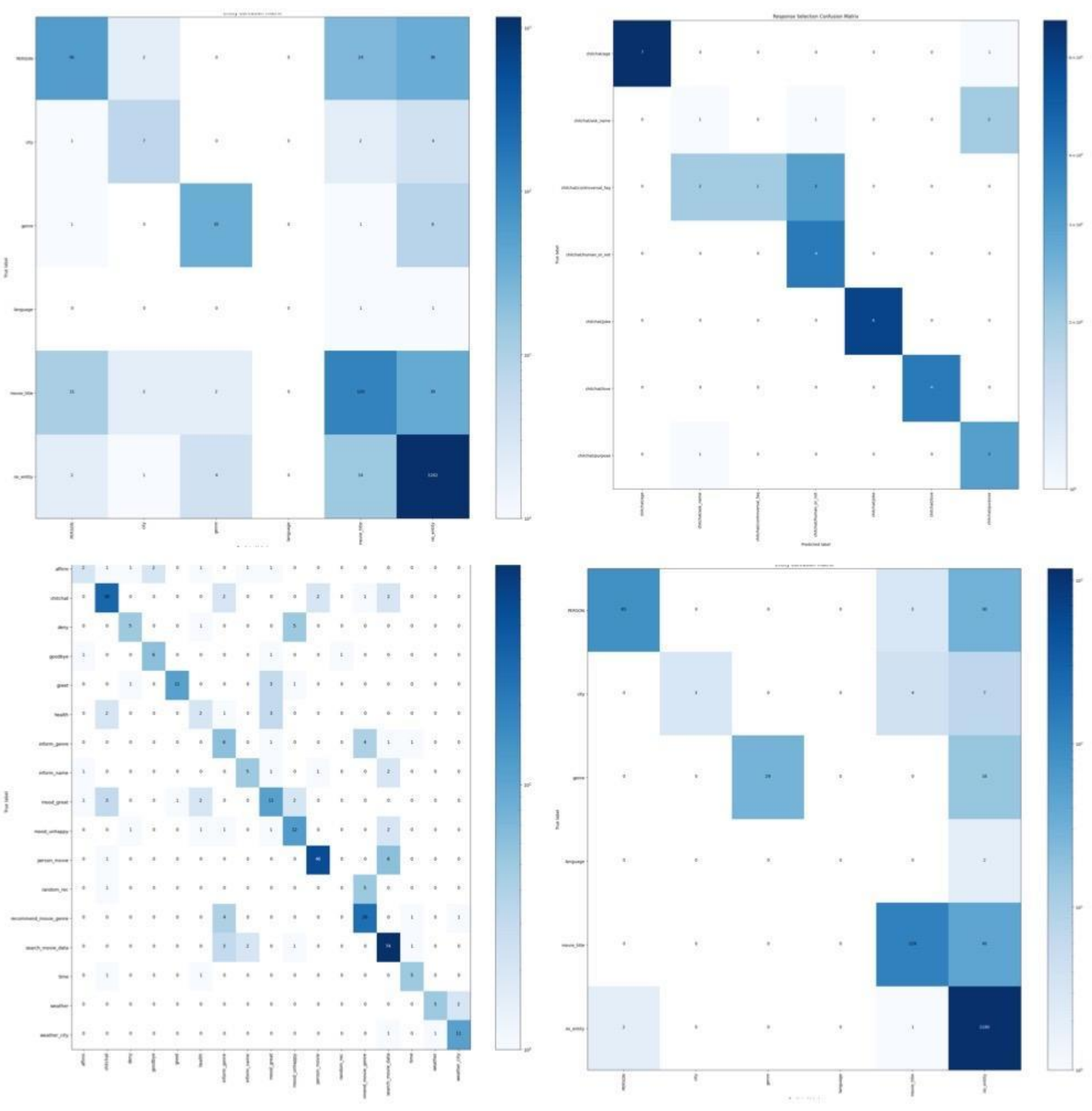


Figure 2: Confusion Matrix

2. **Dialogue Management Component:**
   To get an overview of this part, I decided to put the human aspect here and perform a general evaluation on how good my assistant is. To think about this, I have come up with 5 simple questions for users to understand but also enough to cover the aspect of task completion, fallback policies and how the model adapts the situation. All the answers will be given on the scale of 1-5. This evaluation was done by 15 people (who spent around 5- 20 minutes with the bot).
   - How good do you think the model can make a chit chat conversation with you?
     - Average result : 3.58
   - How close is it to a normal human-human conversation?
     - Average result : 1.67
   - How accurate the model is to follow and answer to the point of your question?

    - Average result: 2.5
- How good the model can answer your movie question?
    - Average result : 1.67
- How good the model can answer your weather question?
    - Average result: 4.5
- How likely do you want to re-use this assistant again?
    - Average Result: 3.5
- In conclusion, what is the final score for this assistant?
    - Average Result: 3.16

VI. Conclusion

In conclusion, despite the fact that the model has not met the expectation to be a good model, this is a perfect opportunity for me to study and understand how to design conversations between humans and machines. Also during the course, people who gave the evaluation also gave me some advice about why the model is not good? What is lacking from the model? This feedback actually also covers the aspect of the Human Computer Interaction part that has been taught at the second lecture of this course.