Language Understanding System: Final Project

Montagner Andrea - 189514
andrea.montagner@studenti.unitn.it

Abstract

This document contains the report of the second project of the course of Language Understanding System. The goal is to develop a simple dialog system within Rasa framework in the movie domain, which is able to interact with the user answering to pertinent questions (*e.g.* director names, actors name, release year etc etc). All necessary steps will be explained, starting from the pre-processing of the data to fit the requests of Rasa, to the configuration of the database and the query system to access information. The preferred language for the both was Python (v2.7), whereas the RDBMS for the database was MySql.

1 Introduction

It has been almost a decade since the trend of bots' development had a sharp increase and many products were sold on the market. Even though many of them had a lot of success, one of the main issue that many both were not able to face was to recognize the context and this derives from the fact that many of them used state machine to solve this task. With the spreading of new techniques, such as machine learning algorithms, a lot of research has been done on this issue, but has not translated into actual developers tool.

Rasa is a framework which provides a new approach to conversational softwares: instead of taking hard-coded rules it exploits the fact that if on one hand understanding when the bot is wrong is easy, on the other hand understanding *why* it is wrong can be very tricky. Following this way, it is possible to decide everything the bot can do or say, even the training can be done either in a supervised way (if data are available) or with an interactive learning starting from scratch.

2 Data Analysis

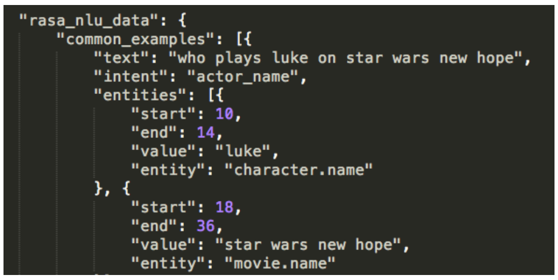
The provided dataset is the same of the previous project, namely the Microsoft NL-SPARQL dataset and it was already split into two parts: one for training and one for testing. In addition, each of these groups of data was divided into main data for the Movie Domain and additional features. The latter contains labels for each sentence.

To summarize, what was given were five files:

- **NL-SPARQL.train.data** containing a two columns set of data, tab separated, where the first column represent the words and the second tags,
- **NLSPARQL.train.utt.labels.txt** containing a single column set of data, identifying the labels for each sentence,
- **NL-SPARQL.test.data** containing a two columns set of data, tab separated, where the first column represent words and the second tags,
- **NLSPARQL.test.utt.labels.txt** containing a single column set of data, identifying the labels for each sentence,
- **moviedb.sql**: the database of the movie domain

2.1 Data pre-processing

The dataset had to be manipulated in order to be compatible with Rasa. A python script (*data_processing.py*) which takes the "raw" data in input and outputs them in a .json file in the correct format. An example of the result is shown in Figure 1.



```
"rasa_nlu_data": {
    "common_examples": [{
        "text": "who plays luke on star wars new hope",
        "intent": "actor_name",
        "entities": [{
            "start": 10,
            "end": 14,
            "value": "luke",
            "entity": "character.name"
        }, {
            "start": 18,
            "end": 36,
            "value": "star wars new hope",
            "entity": "movie.name"
```

Figure 1: Example of the training data after processing

3 Rasa-Core && Rasa-NLU

As stated before, Rasa is a framework that allows to create a complete Dialogue System, meaning training the NLU and creating the dialogue manager to handle the conversation. It is based on some concepts, that will be explained in the next subsections.

3.1 Texts

The text is the search query, namely the entire sentence on which the intent and the entities will be extracted.

In this case, each line of the initial dataset contains a word of a sentence and the end of the sentence is delineated by an empty line. The text is a reshape of the sentence in one line.

3.2 Entities

An Entity consists in specific parts of the text which need to be identified. Each entity is specified with a start and end value that identify respectively the starting and the end point of the entity.

In this particular case, the dataset has been provided with tagged word with IOBs, and consequently the tags are used as entity. Note that IOBs are of three type:

I Inside of span

O Outside of span

B Beginning of span

Only ”B” and ”I” tags are taken into consideration as entities, since ”Outside of span” words do not bring any additional useful information.

3.3 Intents

The intent is the intent that should be associated with the text. Basically, it is what the user is probably looking for in the sentence.

In this case, intents are taken from the label file and from some examples in the Rasa repository on Git.

3.4 Actions

Actions are all the things the bot can actually do. They have to follow a fixed structure, being invoked by calling the *action.run()* method. Actions can do multiple things, like:

- Respond to a user
- Query a database

They have all to extend the *Action* class, and can be of various type. The easiest are called UtterActions, which just send a message to the user. There is also the possibility of creating custom, more complex actions which perform composite commands/queries.

Custom actions are implemented in *actions.py*

3.5 Domain

Finally, the last crucial component is the domain, which defines the universe in which the bot operates. It is a *.yml* file, which contains:

- Slots
- Entities (3.2)
- Intents ((3.3)
- Templates
- Actions (3.4)

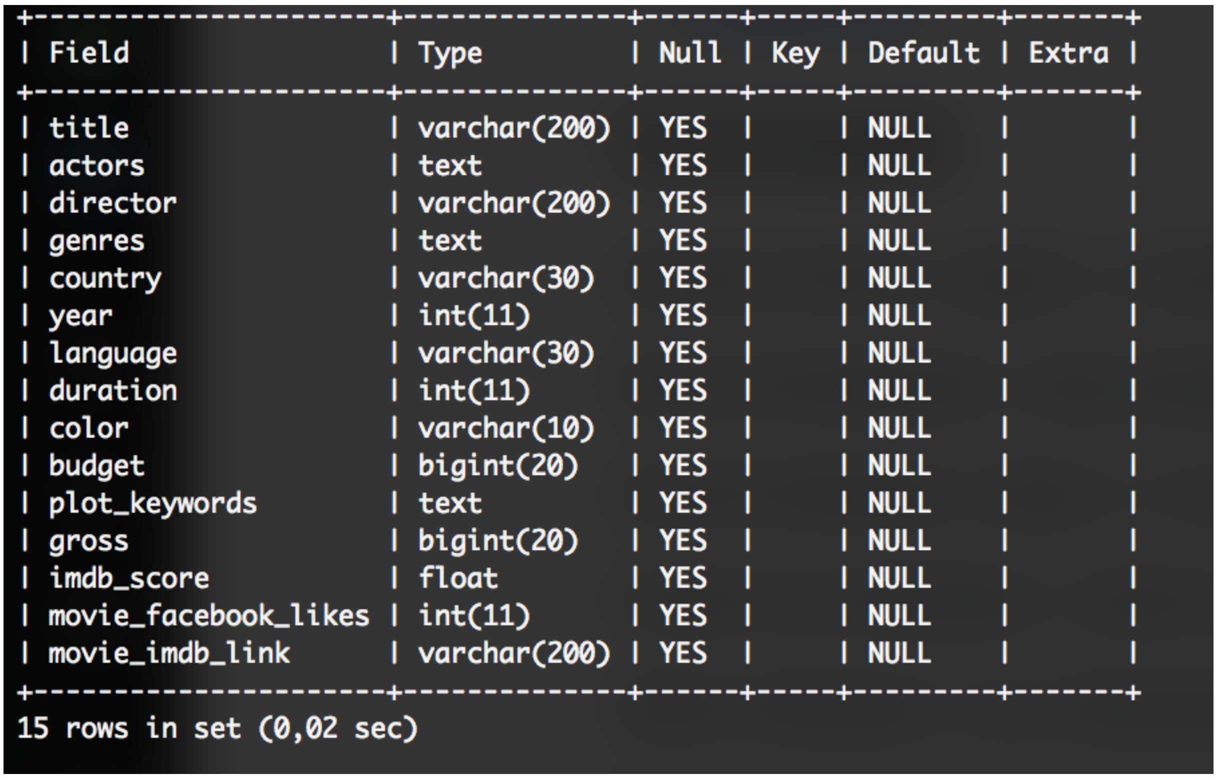
where slots contain what the bot needs to keep track during the conversation, entities, intents and actions are exactly what is explained few subsection above, and templates represent answers to the basic actions, like initial/final greetings, negative result to search and so on.

The domain file is *domain.yml*.

4 Database

4.1 MySql

As far as the database part is concerned, it is pretty simple and straightforward. MySql has been used as RDBMS because it is well interfaced with Python through toolkit like SQLAlchemy (**?**). For this project the file *moviedb.sql* has been loaded into MySql in order to be accessible to queries. It is composed by 2 tables, Movie and Movies, of which only movie has been used. Structure of the table is shown in Figure 2



```
+---------------------+-------------+------+-----+---------+-------+
| Field               | Type        | Null | Key | Default | Extra |
+---------------------+-------------+------+-----+---------+-------+
| title               | varchar(200) | YES |     | NULL    |       |
| actors              | text        | YES  |     | NULL    |       |
| director            | varchar(200) | YES |     | NULL    |       |
| genres              | text        | YES  |     | NULL    |       |
| country             | varchar(30) | YES  |     | NULL    |       |
| year                | int(11)     | YES  |     | NULL    |       |
| language            | varchar(30) | YES  |     | NULL    |       |
| duration            | int(11)     | YES  |     | NULL    |       |
| color               | varchar(10) | YES  |     | NULL    |       |
| budget              | bigint(20)  | YES  |     | NULL    |       |
| plot_keywords       | text        | YES  |     | NULL    |       |
| gross               | bigint(20)  | YES  |     | NULL    |       |
| imdb_score          | float       | YES  |     | NULL    |       |
| movie_facebook_likes | int(11)    | YES  |     | NULL    |       |
| movie_imdb_link     | varchar(200) | YES |     | NULL    |       |
+---------------------+-------------+------+-----+---------+-------+
15 rows in set (0,02 sec)
```

Figure 2: 'Movie' table of the database

4.2 Accessing

In order to access the database and make queries to obtain information, it has been used a Python access tool, named SQLSoup (**?**), which is built on top of SQLAlchemy. An issue when thinking about how to search the database was the fact that target users are very different from each other and can make grammar mistakes in searching information. To solve this problem the *ilike* function has been used, which is the correspondent of the *LIKE* operator of the SQL language, in order to collect all similar matches to the research. The only cons this solution has, is that there is the necessity of another function which finds the best match between the one returned from the search.

The implementation of the database manager can be found in *db_manager.py*.

5 Bot

The bot is implemented in *bot.py* and its structure is very simple, it let the user decide in which way to run depending on the input parameter:

- **train-nlu**

  This is the first step to be done, it takes as inputs the training data and the domain file, then it trains a Trainer over the data and stores the result in a directory.

- **train-dialogue**

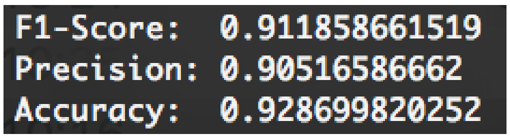
  After training the NLU, it is ready for the question/answer phase, but it does not know how to handle the conversation. For this purpose, a dialogue manager (DM) is trained passing as input the NLU previously created, the policies (Keras and Memoization in this case). the DM is trained over a data file, here identified as *stories.md*, which contains examples of conversation in order to help to understand the flow. There is also the possibility of an "online_training", that is exactly the one previously referenced as *interactive learning* when describing Rasa. In this way, it is possible to train the agent step by step, immediately correcting its answers when wrong.

- **run**

  Standard way to run the bot, once both training steps are done.

6 Training and evaluation

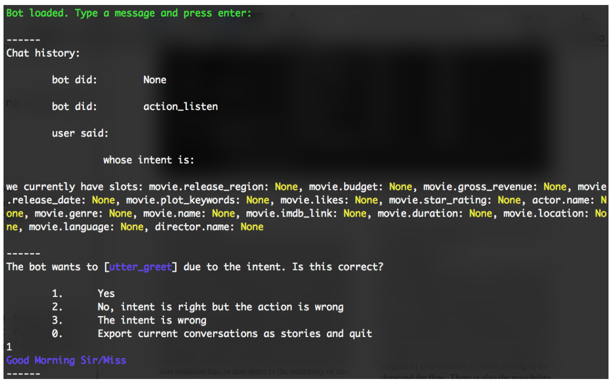
NLU has been trained using Rasa standard evaluation method (*python -m rasa_nlu.evaluate*) and results are above 90% for all criterions: F1 score, precision. accuracy.



```
F1-Score:  0.911858661519
Precision: 0.90516586662
Accuracy:  0.928699820252
```

Figure 3: Results of evaluation of the NLU

The dialogue manager has been trained mainly through with the online training method, a particular procedure where the bot runs in a debug-like fashion. Here it is possible to correct bot's actions step by step creating stories from which it will learn. The structure of this procedure is shown in Figure 4.



```
Bot loaded. Type a message and press enter:

------
Chat history:

        bot did:        None

        bot did:        action_listen

        user said:

                whose intent is:

we currently have slots: movie.release_region: None, movie.budget: None, movie.gross_revenue: None, movie
.release_date: None, movie.plot_keywords: None, movie.likes: None, movie.star_rating: None, actor.name: N
one, movie.genre: None, movie.name: None, movie.imdb_link: None, movie.duration: None, movie.location: No
ne, movie.language: None, director.name: None

------
The bot wants to [utter_greet] due to the intent. Is this correct?

        1.      Yes
        2.      No, intent is right but the action is wrong
        3.      The intent is wrong
        0.      Export current conversations as stories and quit
1
Good Morning Sir/Miss
------
```

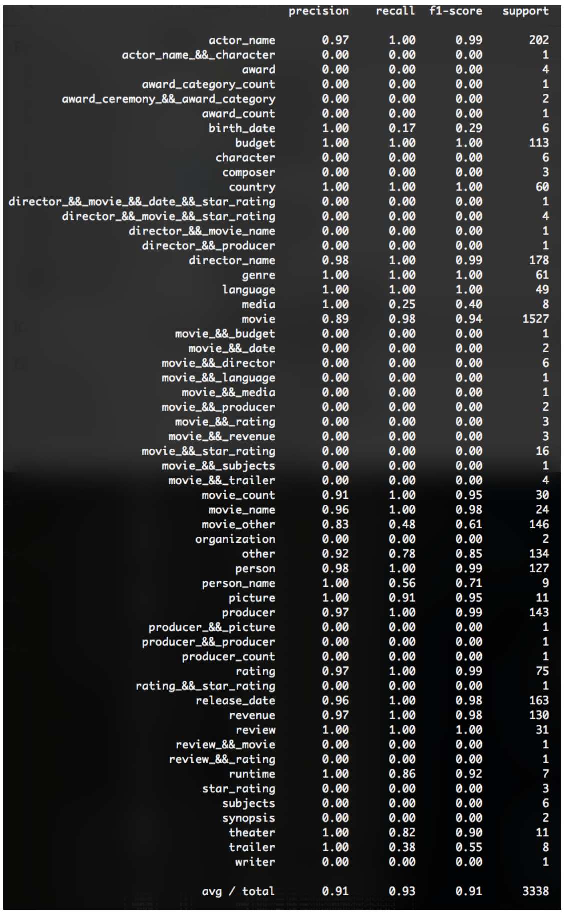
Figure 4: Interactive/online training

The bot is able to answer in a quite good way to almost all simple questions based on a movie title. Director's information, actor's list, release year or country, budget, gross and also the rating of the requested movie on IMDB.

7 Conclusions and Future Work

In this project a very simple implementation of a chat-bot within Rasa framework is shown, even if with some limitations. It is possible indeed to train the NLU with satisfactory results (above 90% of F1) as shown above and also to train the dialogue manager both with the online and the supervised training.

The pipeline follows the implementation provided in the example in the Rasa GitHub repository with some insight from the official documentation.



```
                                            precision    recall  f1-score   support

                               actor_name       0.97      1.00      0.99       202
                   actor_name_&&_character       0.00      0.00      0.00         1
                                    award       0.00      0.00      0.00         4
                     award_category_count       0.00      0.00      0.00         1
         award_ceremony_&&_award_category       0.00      0.00      0.00         2
                              award_count       0.00      0.00      0.00         1
                               birth_date       1.00      0.17      0.29         6
                                   budget       1.00      1.00      1.00       113
                                character       0.00      0.00      0.00         6
                                 composer       0.00      0.00      0.00         3
                                  country       1.00      1.00      1.00        60
director_&&_movie_&&_date_&&_star_rating       0.00      0.00      0.00         1
         director_&&_movie_&&_star_rating       0.00      0.00      0.00         4
                  director_&&_movie_name       0.00      0.00      0.00         1
                    director_&&_producer       0.00      0.00      0.00         1
                            director_name       0.98      1.00      0.99       178
                                    genre       1.00      1.00      1.00        61
                                 language       1.00      1.00      1.00        49
                                    media       1.00      0.25      0.40         8
                                    movie       0.89      0.98      0.94      1527
                         movie_&&_budget       0.00      0.00      0.00         1
                           movie_&&_date       0.00      0.00      0.00         2
                       movie_&&_director       0.00      0.00      0.00         6
                       movie_&&_language       0.00      0.00      0.00         1
                          movie_&&_media       0.00      0.00      0.00         1
                       movie_&&_producer       0.00      0.00      0.00         2
                         movie_&&_rating       0.00      0.00      0.00         3
                        movie_&&_revenue       0.00      0.00      0.00         3
                    movie_&&_star_rating       0.00      0.00      0.00        16
                       movie_&&_subjects       0.00      0.00      0.00         1
                        movie_&&_trailer       0.00      0.00      0.00         4
                              movie_count       0.91      1.00      0.95        30
                               movie_name       0.96      1.00      0.98        24
                              movie_other       0.83      0.48      0.61       146
                             organization       0.00      0.00      0.00         2
                                    other       0.92      0.78      0.85       134
                                   person       0.98      1.00      0.99       127
                              person_name       1.00      0.56      0.71         9
                                  picture       1.00      0.91      0.95        11
                                 producer       0.97      1.00      0.99       143
                     producer_&&_picture       0.00      0.00      0.00         1
                    producer_&&_producer       0.00      0.00      0.00         1
                           producer_count       0.00      0.00      0.00         1
                                   rating       0.97      1.00      0.99        75
                   rating_&&_star_rating       0.00      0.00      0.00         1
                             release_date       0.96      1.00      0.98       163
                                  revenue       0.97      1.00      0.98       130
                                   review       1.00      1.00      1.00        31
                         review_&&_movie       0.00      0.00      0.00         1
                        review_&&_rating       0.00      0.00      0.00         1
                                  runtime       1.00      0.86      0.92         7
                              star_rating       0.00      0.00      0.00         3
                                 subjects       0.00      0.00      0.00         6
                                 synopsis       0.00      0.00      0.00         2
                                  theater       1.00      0.82      0.90        11
                                  trailer       1.00      0.38      0.55         8
                                   writer       0.00      0.00      0.00         1

                              avg / total       0.91      0.93      0.91      3338
```

Figure 5: Classification report of NLU training with Rasa

I personally really liked this topic, in particular related with the movie domain. For this reason, I think that one of the future improvements this bot will have is use Rasa custom policies to model more complex queries, like "Group me all movies with actor X" or "Tell me how many times actor X is involved in a Thriller movie". Moreover I would also like to connect it to chatting platform like telegram in order to have it mobile for any need.