

Rasa Conversational AI - Flight Booking Chatbot
Human-Machine Dialogue Course
University of Trento
Master of Artificial Intelligence Systems

Ali AKAY

Instructors: Prof. Giuseppe Riccardi

1 Introduction

Rasa Conversational AI assistant is based on natural conversations that how humans interact with each other by considering what earlier the context was sent and what actions are to be taken in reference to the contexts and gracefully handling the unexpected conversation and driving the conversation when the user drifts from normal conversation path and also improve over time thus its far beyond the FAQ Interactions.

1.1 Model type

Conversational chatbot bot has been modelled for task-based type chatbot that helps users accomplish specific tasks may be referred to as task oriented dialogue agents.

In this project, the chatbot is a reservation agent that helps users to book a flight. Agent will lead the conversation and in the end it will book your flight based on the information given by the customer.

1.2 Domain

The chatbot can handle three type of tasks.

1.2.1 Booking Flight Ticket

Based on the nature of the interaction, there are some possible scenarios to ask/text the chatbot for booking a flight ticket. For that reason, the chatbot is modelled for 7 different vary of texting for booking a flight. Here you can see the text by user and explanation.

Rasa forms is used to ask questions for booking flight to users. In total, we have

4 slot for the booking form. Based on user input, required booking form is activated.

- *"Book me a flight" or "Flight ticket"*

In this case, there no given booking information by the user. So, the chatbot have to lead the conversation while asking questions in order to book a flight.

- *"Book me a flight ticket to Milano"*

Here, user only give information about destination city so that, the chatbot understand that **destination** city is given and ask questions in order to fill the blank part in order to book a flight.

- *"Book me a flight ticket to Milano for 2 people"*

User can give **destination** and **number of people** information to the chatbot.

- *"Book me a flight ticket from Milano to Istanbul"*

User can give **departure** and **destination** information to the chatbot.

- *"Book me a flight ticket from Milano to Istanbul for 2 people"*

User can give **departure**, **destination** and **number of people** information to the chatbot.

- *"Book me a flight ticket from Milano to Istanbul between July 15-25"* User can give **departure**, **destination** and **dates** information to the chatbot.

- *"Book me a flight ticket from Milano to Istanbul between July 15-25 for 2 people"* User can give **departure**, **destination**, **dates** and **number of people** information which is all required information to the chatbot.

1.2.2 Airlines and Airport Information

User also can text for short question-answering type for airport and airlines informations.

- *What does the airline code THY stand for?*
- *Where is Gatwick Airport located?*
- *Where is mco airport?*

Answers of these questions always same way based the airline and airport information from database.

1.2.3 Chitchat

FAQs and chitchat are two cases where the chatbot responds with a fixed set of messages, and the assistant should always answer the same way, no matter what has happened previously in the conversation.

1.3 User Group

User group of the conversational AI is very specific people who would like to book a flight. For the future, Train and bus could be added the conversational AI and extend the user group.

2 Conversation Design

Conversational Design is a type of language based on human conversation and experiences. The more natural and enjoyable the interactions, the fewer aversion customers will have to a chatbot.

In this project, the conversational AI is designed mixed initiative type which is agent will lead and ask the questions based on the task as well as agent offers different options based on the current situation and user can choose the way of the conversation.

```
Your input -> book me a flight from Istanbul to Berlin for tomorrow
For how many people should I book? Please say like '5 people'
Your input -> 2
I am going to search flight from Istanbul to Berlin on 2022-06-15 for 2 people, can you please confirm that? Or just say
'y' 'Start again' for new query.
Your input -> yes
Please wait while I am looking for best flight for you!
Here is the 'Best' option based on duration and price. From Istanbul to Berlin for 186 USD at 2022-06-15 -> 10:10 am
- 12:10 pm Pegasus Airlines nonstop 3h 00m SAW - BER
I am going to book it for you. Can you please confirm that? There is also cheapest flights, if you would like to see them
just say 'go for cheapest one'
Your input -> go for cheap one
Here is the cheapest option based on price. From Istanbul to Berlin for 168 USD at 2022-06-15 -> 9:25 am - 11:20 am
Anadolujet nonstop 2h 55m SAW - BER
I am going to book it for you. Can you please confirm that?
Your input ->
```

Figure 1: Offer another option

Since it is a program and not human, it has its limitations which is natural. When it doesn't have the answer, it's better to say that the conversational AI can say don't know than to try to answer and leave the user dissatisfied. In the project, the chatbot says *"Sorry I didn't get that. Can you rephrase?"* when it happens and force the user to text it in the correct way, if it happens many times, the system restarts itself and starts again the conversation.

2.1 Data Description and Analysis

Dataset is created from the same structure and taking as an example from Airline Travel Information System (ATIS) dataset which is a standard benchmark dataset widely used as an intent classification and conversational AI problems. ATIS dataset provides large number of messages and their associated intents that can be used in training a classifier. Within a chatbot, intent refers to the goal the customer has in mind when typing in a question or comment. While entity refers to the modifier the customer uses to describe their issue, the intent is what they really mean.

2.2 Intent and Entities

31 different intent is used in the training dataset. There are 8 entities which are **GPE(Departure, Destination), time, number, email, airport code, airline code, airport name**. For GPE entity, **spacy** is used while for time, number and email **Duckling** is used.

2.3 Slots and Rules

RASA forms is used to ask and get answer from user in systematic way. As it has already mentioned in section 1.2.1, there are 6 different form for 4 slot values which is departure, destination, number of people and date-time. Based on the text by user, right form is activated.

```
Current slots:
  email: aliakay@gmail.com, no_of_people: 2, airline_code: None, airport_name: None, airport_code: None, departure: Istanbul, destination: Roma, time: ('to': '2022-07-26T00:00:00.000+02:00', 'from': '2022-07-15T00:00:00.000+02:00'), requested_slot: None, session_started_metadata: None
```

Figure 2: Example of filled slots

In addition, there are also slots for airport information and airline informations. There are 18 rules for different scenarios such as activating the right form, restarting session, greeting, goodbye, chitchat, out of scope.

2.4 Validation of User Inputs

User inputs also checked whether there is a typo or departure and destination cities can not be same. Usually, RASA form handles this issue but sometimes, I realized that it can be missed. This problem is handle by writing a function in `action.py`

For date/time validation, duckling works perfectly for it, even user texts the dates from past, it gives the next year same date which help to the chatbot. After RASA form is filled, the chatbot asks to user to validate his/her input again to continue to process which help the chatbot to go in right way.

```
Your input -> hi
Hi, how are you today? I'm an AI travel assistant. Here to help you plan the perfect next trip! I can book a flight for you and give information about airports and cities.
Your input -> I want you to book me a flight
Let's start with departure, what is the departure city? Please answer like: 'From Istanbul'
Your input -> from roma
What's your destination? Please answer like: 'To Edinburgh' Here is some hot destinations. [Edinburgh|London|San Francisco|Milano]
Your input -> to Roma
For how many people should I book? Please say like '5 people'
Your input -> 2 adult people
When would you like to book the flight? If it is round trip please say dates like 'July 15-25' or 'July 15 - August 20' otherwise please type your date.
Your input -> tomorrow
Departure and Destination cities can't be same. Please start again.
Let's start with departure, what is the departure city? Please answer like: 'From Istanbul'
```

Figure 3: Validation of wrong input by user

```

Your input -> hi
hi, how are you today? I'm an AI travel assistant. Here to help you plan the perfect next trip! I can book a flight for you and give information about airports and cities.
Your input -> book me a flight ticket
Let's start with departure, what is the departure city? Please answer like: 'From Istanbul'
Your input -> from Roma
What's your destination? Please answer like: 'To Edinburgh' Here is some hot destinations, |Edinburgh|London|San Francisco|Milano|
Your input -> to London
For how many people should I book? Please say like '5 people'
Your input -> 2 people
When would you like to book the flight? If it is round trip please say dates like 'July 15-25' or 'July 15 - August 20' otherwise please type your date.
Your input -> July 15-25
I am going to search flight from Roma to London on 2022-07-15 - 2022-07-25 for 2 people, can you please confirm that? Or just say 'Start again' for new query.
Your input -> yes

```

Figure 4: Validation of good input by user

2.5 Domain Data

Flight informations are taken from kayak dot com website using web scrapping library Selenium and BeautifulSoup in python.

2.6 Conversation Model

Conversational model is built with RASA Framework which is contextual AI assistants and chatbots in text and voice with our open source machine learning framework.

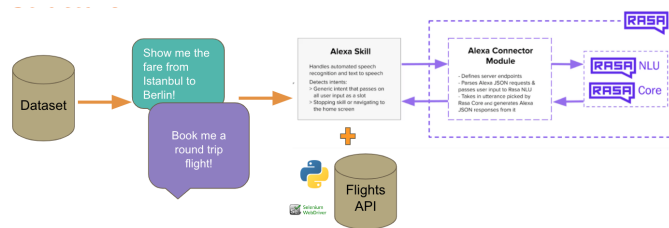


Figure 5: Structure of Conversational Model

For natural language understanding model(NLU), different classifiers, tokenizer, featurizer and extractor is used.

- **Whitespace Tokenizer** is a tokenizer that splits on and discards only whitespace characters.
- **Lexical Syntactic Featurizer** is a component meant to create features that are useful when detecting entities.
- **Count Vectors Featurizers** only adds one feature for each word in your training data.
- **DIET Classifier** is transformer architecture that can handle both intent classification and entity recognition together.

- **Fallback Classifier** steps in when the Intent Classifier is not confident about the intent.
- **Spacy Entity Extractor** is used to extract departure and destination cities using GPE tag.
- **Duckling Entity Extractor** is used for email,numbers and especially date/time.

2.6.1 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.TED for short, is just one of the dialogue policies used by Rasa Open Source to select which action the assistant should take next.

- Rule policy

This policy to force the model to follow a fixed behavior which has been declared inside the rules file.

2.6.2 Vocal Model

Rasa conversational assistant is trained and connected to Alexa, Amazon's cloud-based voice service. This will let users interact with the chatbot either by voice or text through Alexa-integrated devices.Alexa skills handle the voice side of things while Rasa will take care of figuring out conversational AI part.

3 Evaluation

3.1 Feedback from User

After the service is done by chatbot,feedback is asked to user to evaluate the chatbot based on the service has given by the chatbot.

Here you can see the example of it.

```

Your input -> book me a flight from Istanbul to Berlin tomorrow
for how many people should I book? Please say like '5 people'
Your input -> 2 people
I am going to search flight from Istanbul to Berlin on 2022-06-15 for 2 people, can you please confirm that? Or just say 'Start again' for new query.
Your input -> yes
Please wait while I am looking for best flight for you!
Here is the best option based on duration and price.From Istanbul to Berlin for 186 USD at 2022-06-15 ----> 10:10 am - 12:10 pm Pegasus Airlines nonstop 3h 00m SAW - BER
I am going to book it for you.Can you please confirm that? There is also cheapest flights,if you would like to see them ,just say 'go for cheapest one'
Your input -> confirm
could you please share me your email adress to send your flight information?
Your input -> aliakay8@gmail.com
I have sent your flight information to, aliakay8@gmail.com, adress.Have a nice Trip.Do you need anything else?
Your input -> no thanks
Would you like to rate me? Just say 'Bye' or tell me your experience 'Very good' - 'Good' - 'Bad' - 'Very Bad'
Your input -> Very good
Oh, I am so happy to help you! See you next time.

```

Figure 6: User feedback to the chatbot

3.2 Natural Language Understanding

Rasa Open Source lets us validate and test dialogues end-to-end by running through test stories. Firstly, train and test data is generated which has also test stories. After that data and stories validated and tested.

General metrics and actionable feedback on intents. You can see at a glance how your model performs and where it fails: all failing utterances are listed alongside their actual vs. predicted intent in the part of appendix. Also you can find the results in results folder.

3.3 Dialogue Model

Here, test stories are evaluated with trained dialogue model on a set of test stories and there is no error founded. You can see the result in [Figure 9].

3.4 Dialogue Management components

For this part evaluation, human evaluation survey is prepared and asked to 20 people who used and spent 5-10 min with the chatbot. Question is prepared to understand the quality of fallback policies, response selection/generation, task completion and feeling.

Here you can see the questions and results in [Table 1].

Table 1: Human Evaluation for the Conversational AI

Questions	Overall Result %
Using chatbot was fun.	90
I felt comfortable using the chatbot to answer question.	85
How good the chatbot can answer question for booking a flight ticket?	80
Can you use the chatbot for booking a flight ticket?	80
I felt the chatbot leads the conversation.	80
How accurate the responses of the chatbot.	70
If you get lost in the conversation how likely chatbot is recover the conversation?	30
How close is it to a normal human-human conversation?	30
I felt comfortable using the chatbot to ask question.	20
What is your overall score for this experience?	70

4 Conclusion

As you can see in the human evaluation, the conversational AI of the project is feel like more computer than feel natural because when the chatbot asks questions, expected answer type is also given to the user to text same type answer in order to keep conversation alive. Even though the model is enough for the requirement, conversational AI part of the project is open for development to get more robust and more nature conversation between user and the chatbot. Also, mission of the chatbot can be extent easily for booking for train, bus, hostel which makes it a travel assistant.

5 Appendix

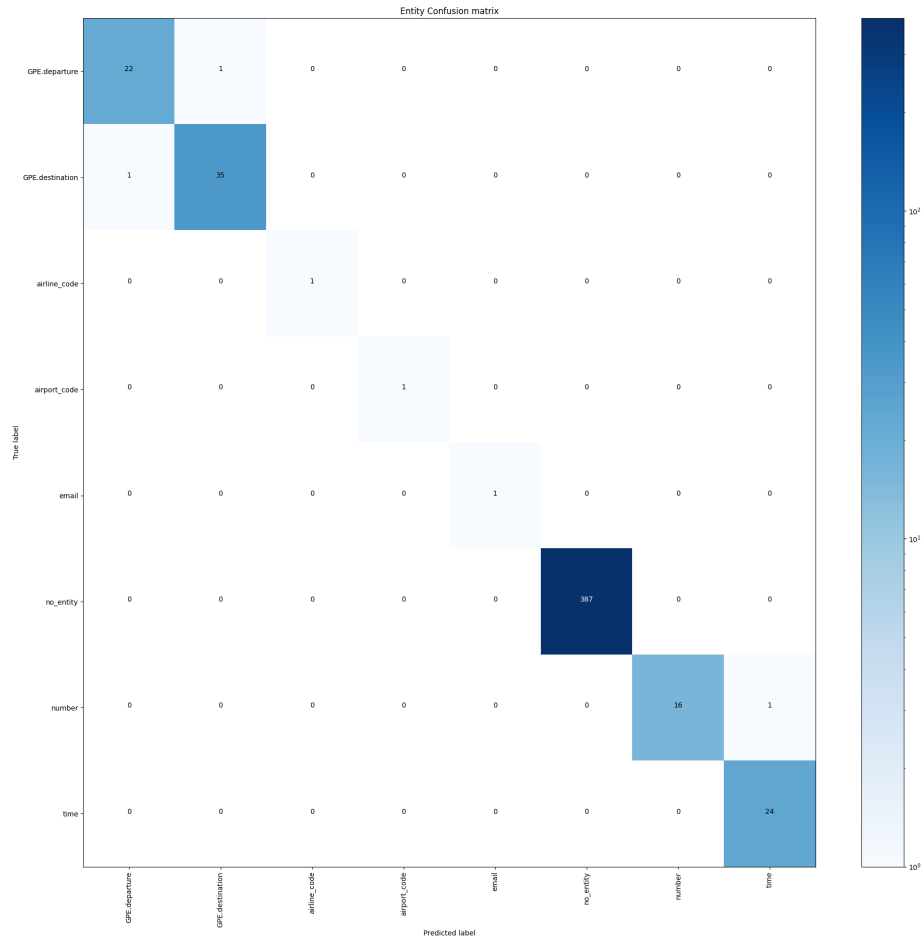


Figure 7: DIET Classifier Confusion Matrix

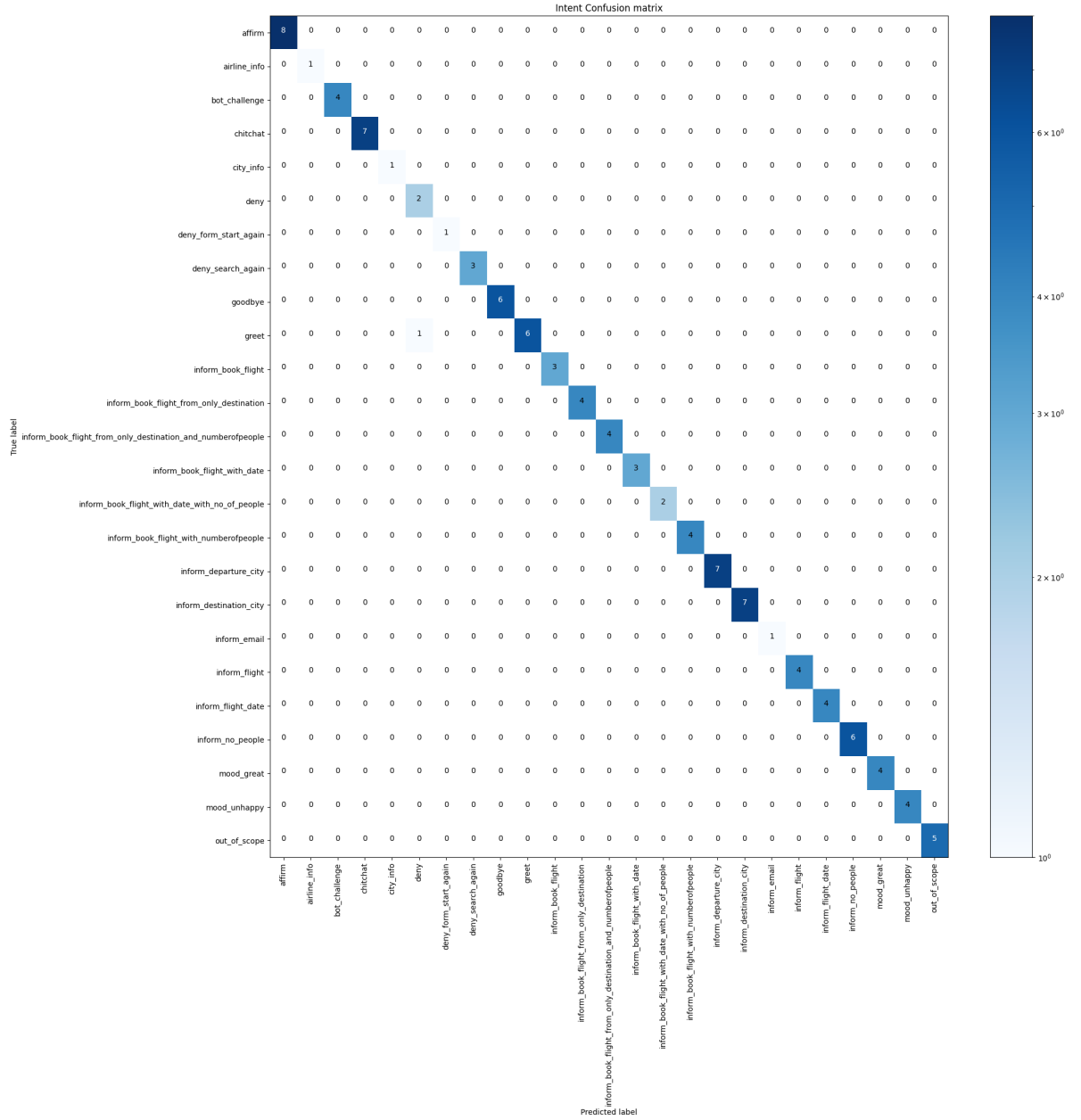


Figure 8: Intent Confusion Matrix

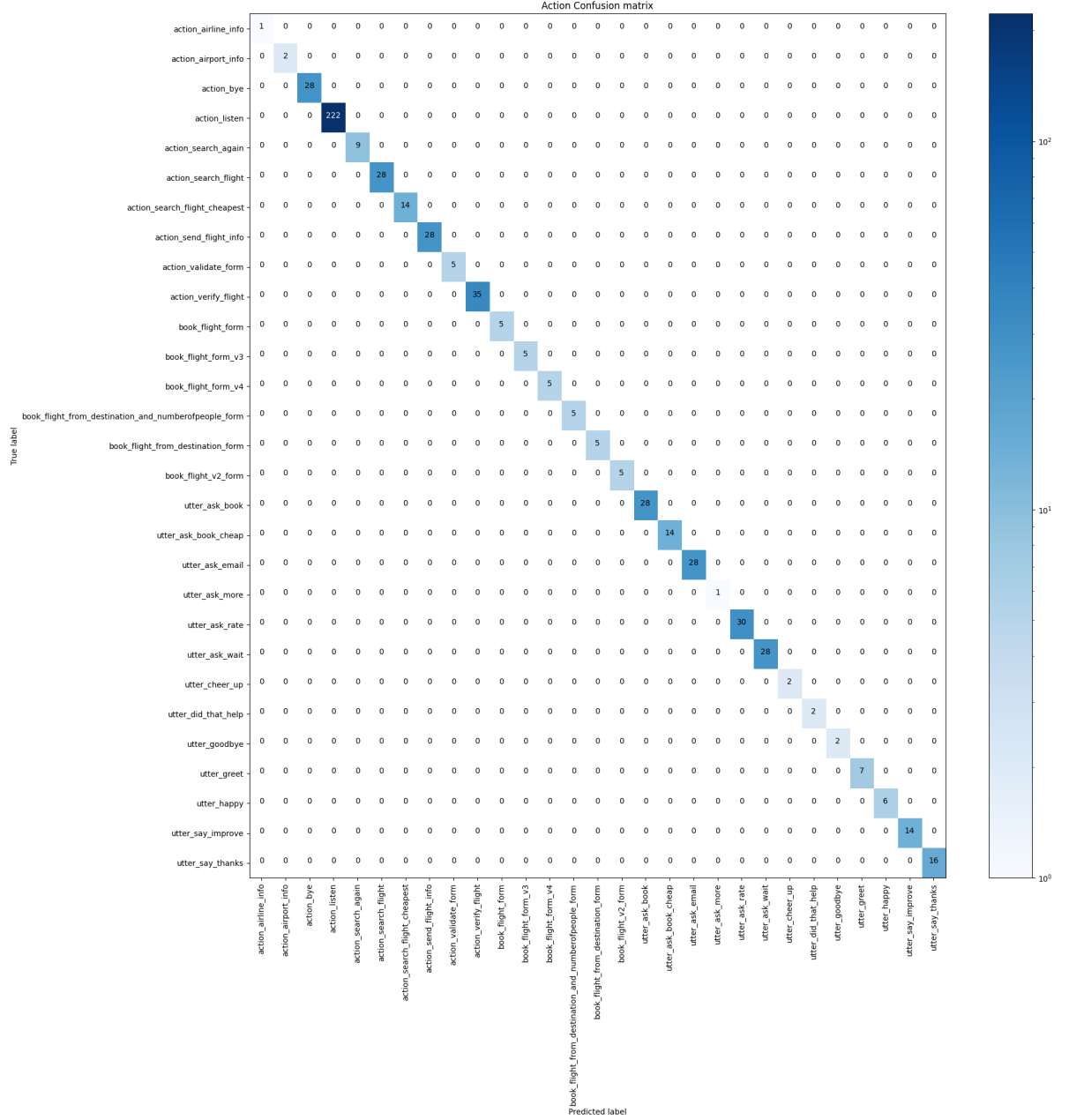


Figure 9: Story Confusion Matrix