Hotel booking skill for Alexa

Michela Lorandi - 207522
michela.lorandi@studenti.unitn.it

Abstract

The goal of this report is to describe the Hotel booking skill for Alexa. This skill has the objective to help the user while searching and booking a hotel.

1 Introduction

The hotel booking skill for Alexa has the goal to find and book hotels based on the user needs. The skill can mainly execute two operations:

1. Search hotels in a given zone;
2. Book a specified hotel.

The code is available on github at https://github.com/michiL96/hotel_booking_skill

2 System description

The system can mainly execute two tasks: search hotels in a given zone and book a specified hotel. For the execution of the tasks, the system takes some assumptions that simplify the execution flow, those assumptions are:

1. All the rooms are available everyday;
2. There is not a maximum number of people per room;
3. There is not a maximum number of rooms per hotel.

The system also contains a method that has the goal to manage the response of the user when the user doesn't know the response of the question requested by the system. In order to do this, the system takes some decisions based on what was requested. More details will be described in the sections below (Section 2.1 and Section 2.2).

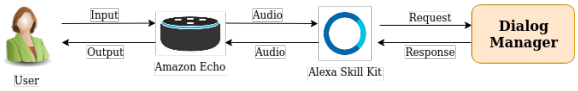

Figure 1: The flow of the skill: the user speak to Amazon Echo with Alexa; the Echo sends the audio to the Alexa Skill Kit that is the NLU; the request is sent to the Dialog Manager that processes it; the response generated by the Dialog Manager is sent to the Alexa Skill Kit that is the NLG; the resulting audio is sent to Amazon Echo; the latter communicates the reply to the user.

Furthermore, there is a mechanism of disambiguation for the information provided by the user. For example, both the fields number of people and number of rooms are a numerical value and when the system asks the value of one field, we need to be able to disambiguate between the two fields and know which one we are looking for. For this reason, inside the dialog state there is a variable that contains the name of the requested field.

Finally, the system has a method that allow the user to select a hotel from the presented list in order to book it. The selection method is also used to select a city from a list of cities when multiple cities with the same name are found. In the latter case, the selection mechanism is used to disambiguate the correct city.

2.1 Booking task

The booking task has the goal to book the specified hotel. In order to properly execute this task, the system needs some required fields, that are: hotel name; city; arrival date; departure date; number of people; number of rooms. If the user wants, he can specify the stars, the address and the region of the hotel. If the user doesn't specify a required field, the system request the field and wait it in order to continue the execution flow.

After the hotel name is specified, the system checks if the hotel is present in the database. If

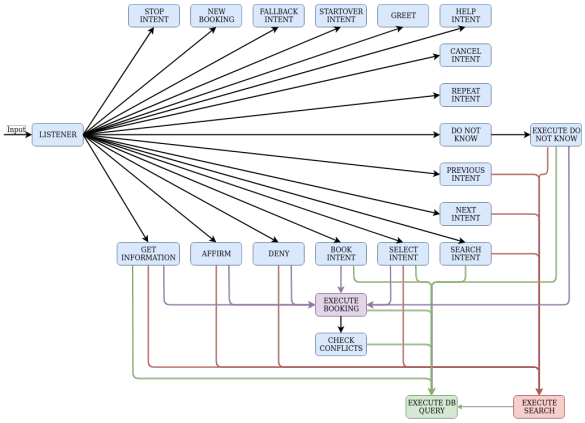

Figure 2: The representation of the Dialog Manager.

the hotel is not present in the database, the system tells the user that it's not possible to book that hotel, otherwise it can continue booking. Furthermore, if the city field is not specified, the system search the specified hotel in the database in order to retrieve its city and ask the user if the found city is the correct city.

When all the required information are obtained, the system executes a method that checks the presence of conflicts inside the given information. The possible conflicts are:

1. the number of rooms is greater than the number of people;
2. the arrival date is after the departure date;
3. the arrival date is in the past;
4. the departure date is in the past;
5. the stars provided by the user are different from the number of stars in the database;
6. the location information (city, address) are not consistent with the hotel information contained in the database.

After the resolution of all conflicts, the skill can book the hotel and tell to the user the occurred booking.

As explained above in Section 2, the system can manage when the user doesn't know a response. In details, in the booking task there are the following cases:

- resolving the conflict regarding the number of rooms and the number of people, the system decides to keep the number of people;
- resolving the conflict regarding the arrival date and the departure date, the system decides to swap the two dates in order to resolve the conflict;
- resolving the conflict regarding city, stars and address, the system assumes the user input is underspecified, hence the reference is the value in the database;
- the requested field is the arrival date or the departure date, the system tells the user that the field is mandatory in order to continue and book the hotel;
- the requested field is the number of room or the number of people, the system assumes that the value is 1 and continues with the booking;
- the requested field is the city, the system assumes that the city prompted to the user is the correct city and continues with the booking;
- the requested field is the hotel name, the system proposes to the user to execute a search and find some hotels.

One drawback of this task is that the system is able to identify the hotels only if the user specifies the complete correct name. If the user specifies the partial hotel name, the system is not able to correctly choose the hotel. Furthermore, the system is able to book only the hotels contained in the database. If a hotel is not present in the database, the system doesn't execute the booking.

2.2 Search task

The search task is a little bit more complex than the booking task because it involves the use of some filters in order to retrieve hotels that satisfy the user's needs. For the search task there is only one required field: the city. This task retrieves the hotels present in a specified city and refines the search by using some additional filters: the address near the hotels you want to look at (with a radius of 2km) and the hotel's stars. The hotels are presented three per times introducing a pagination method. The user can navigate between the pages by asking to go back and forth. Furthermore, the hotels are ranked based on increasing price when the address field is blank, while when the address field is specified, the hotels are ranked based on the distance from the address and on the price.

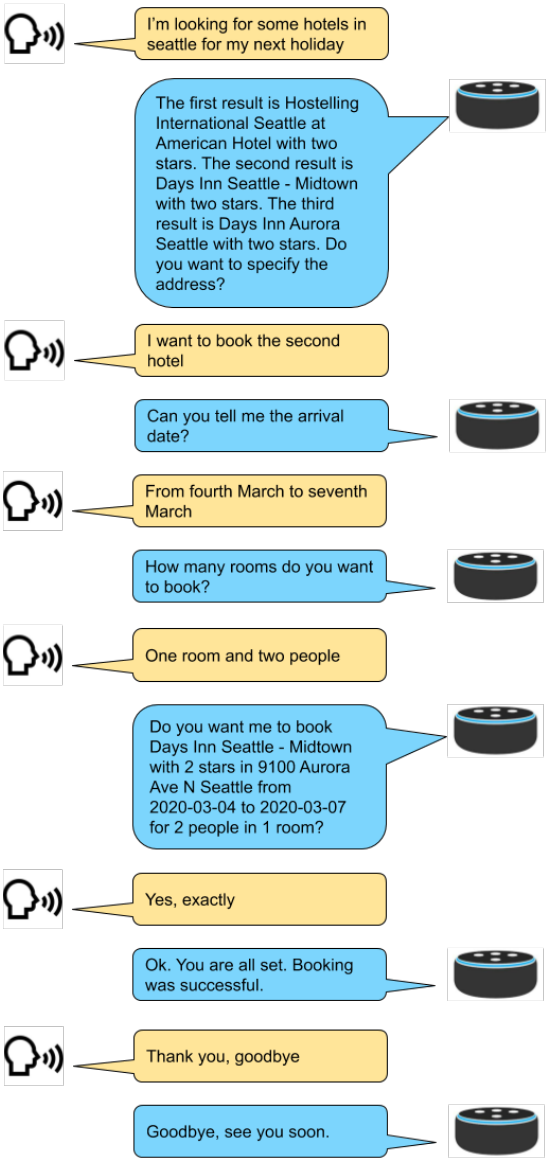

Figure 3: An example of interaction with the system.

The system can also perform a random search, this means that if the user is not sure about where he wants to go, he can search for hotels and the system randomly choose the destination city.

Furthermore, once the system provides to the user a list of possible hotels, the user can simply book one hotel by saying the ordinal number of the hotel in the list, for example "I want to book the second hotel".

As explained above in Section 2, the system can manage when the user doesn't know a response. In details, in the searching task there are the following cases:

- resolving the selection of the correct city when multiple cities with the same name are found, the system select the first city with the specified name;
- the user doesn't know the answer to the request about obtaining more result, the system assumes that the user doesn't want to have more results and asks if the user wants to book a hotel;
- the user doesn't know the answer to the request about booking a hotel, the system assumes that the user doesn't want to book a hotel and asks for a new task;
- the requested field is the city, the system randomly chooses a city in the database and continues with the search task;
- the requested field is the address, the system skips this question and continues asking if the user wants to specify the number of stars;
- the requested field is the stars, the system skips this question and continues asking if the user wants to obtain more results.

2.3 Database

The skill accesses a database to retrieve all the hotels information. The database contains 65 692 hotels located in USA and UK. The dataset used to populate the database was found on github[1].

The database contains only one table: HOTELS. This table contains the following attributes: id, hotelName, stars, price, cityName, countryCode, countryName, address, location, url, latitude, longitude.

[1] https://github.com/lucasmonteiro001/free-world-hotel-database

2.4 Intents

The system is composed of many intents, some custom intents and some built-in intents. The custom intents are:

1. book_hotel: used to book a hotel;
2. search_hotel: used to search hotels;
3. get_information: used to specify additional information;
4. do_not_know: used when the user doesn't know the answer of the question;
5. selection: used to select one hotel in the provided list and book it or to select one city in order to disambiguate between cities with the same name.

The built-in intents used in the skill are: AMAZON.YesIntent; AMAZON.NoIntent; AMAZON.FallbackIntent; AMAZON.CancelIntent; AMAZON.HelpIntent; AMAZON.StopIntent; AMAZON.StartOverIntent; AMAZON.PreviousIntent; AMAZON.NextIntent; AMAZON.RepeatIntent. The goal of each of these intents is described in the Amazon documentation[2].

The intents can contain some slots with the goal to capture the information given by the user. The slots, with the relative slot types, of each intent can be seen in the Table 4.

2.5 Utterances

The training dataset has been manually built, because there isn't an available dataset that addresses the tasks of this project. As a baseline, a restaurant booking dataset has been taken into account and each utterance has been adapted to the hotel booking domain. The Table 1 shows the number of utterances present in each custom intent, i.e. the intents not built-in in Amazon.

3 Experiments

For the evaluation of the system, the skill was used by 3 volunteers, each one executing the test twice. A task was considered successful if the system correctly understood what the user was saying and if the correct intent was selected by the system.

[2] https://developer.amazon.com/en-US/docs/alexa/custom-skills/standard-built-in-intents.html

Intent name	Number of utterances
search_intent	290
book_intent	34
get_information	207
selection	112
do_not_know	7

Table 1: In this table, we can see the number of utterances contained in each custom intent of the system.

Intent name	Number of utterances
StopIntent	27
YesIntent	36
NoIntent	17
NextIntent	30
PreviousIntent	28

Table 2: In this table, we can see the number of utterances added in the Amazon built-in intents.

In order to measure the task success, Accuracy, Precision, Recall and F1 measure have been used, as shown in Table 3. The Accuracy is calculated by the utterances correctly identified by the system divided by the total number of utterances. The Precision is calculated considering the utterances predicted to be a certain intent and checking how many are actually labeled with that intent. The Recall is calculated considering the utterances labeled with a certain intent and checking how many of them are correctly identified by the system. The F1 measure is the harmonic average between Precision and Recall. All the metrics shown are calculated taking the average score of each test. These metrics are also calculated based on slots identified in each utterance.

After the tests, the users also responded to the question "How much do you feel the system understood you?" scoring the response in a scale between 1 and 5. The average result is 3.5 point out of 5. This means that the users, in general, feel that the system understand them but not completely. In fact, the system has some problems in recognizing the correct intents in some specific situation, for example when the user wants to book an hotel after the execution of a search. This is due to the fact that there are too few utterances for the booking intents. Furthermore, the selection of the hotel after the search can be done in many ways. This problem may be resolved, or at least smoothed, by adding more diverse utterances. Another situation in which the system has some prob-

	A	P	R	F1
I	0.84	0.79	0.81	0.80
S	0.84	0.90	0.82	0.85

Table 3: The task success calculating Accuracy (A), Precision (P), Recall (R) and F1 measure. The metrics are calculated based on the intents (I) and on the slots identified in the sentences (S).

lems is when the user specifies additional information. For example, sometimes the identification of AMAZON.Date and AMAZON.Ordinal is not correct, causing the system to misunderstand the user and take the wrong action. In details, sometimes AMAZON.Ordinal takes a non-numerical value, the consequence is that the system cannot correctly process the request.

4 Conclusions

In conclusions, the system is able to search and book hotels in different cases, but there can be done some improvements. First of all, adding more real utterances can help the system in better recognize the correct intents and satisfy the user needs. The utterances dataset used was built using only my knowledge, for this reason it can be limited in some aspects because the structure of the sentences are similar and use few words. Furthermore, a possible improvements is to let the user specify more detailed information, for example specify the number of people for each room.

The database quality is not high, in fact there are some errors in the city names. Improving the quality of the database will lead to a better user experience with the hotel booking skill and will help the system in better satisfy the user needs.

Another thing that can be improved is the search filtering. At the current state, the available filters are the address and the number of stars, a filter that can be added is the price. Furthermore, using the address filter, the system search hotels using a fixed radius of 2Km, an improvement can be the introduction of a radius selected by the user.

A Appendices

The Table 4 shows the slots contained in each custom intent of the skill. For each slot is specified its type.

The slot type *hotel_name* is a custom slot type that contains the list of hotel names contained in the database.

Table 4: In the table are shown the intents with the relative slots and slot types.

Intent name	Slot name	Slot type
book_hotel	hotel_name	hotel_name
	city	AMAZON City
	region	AMAZON Region
	address	AMAZON PostalAddress
	arrival_date	AMAZON DATE
	departure_date	AMAZON DATE
	number_room	AMAZON NUMBER
	number_people	AMAZON NUMBER
	stars	AMAZON NUMBER
search_hotel	city	AMAZON City
	region	AMAZON Region
	address	AMAZON PostalAddress
	arrival_date	AMAZON DATE
	departure_date	AMAZON DATE
	number_room	AMAZON NUMBER
	number_people	AMAZON NUMBER
	stars	AMAZON NUMBER
selection	ordinal_number	AMAZON Ordinal

Continued on next column

Continued from previous column

Intent name	Slot name	Slot type
	hotel_name	hotel_name
get_information	hotel_name	hotel_name
	city	AMAZON City
	region	AMAZON Region
	address	AMAZON PostalAddress
	arrival_date	AMAZON DATE
	departure_date	AMAZON DATE
	number_room	AMAZON NUMBER
	number_people	AMAZON NUMBER
	stars	AMAZON NUMBER
	number	AMAZON NUMBER
	date	AMAZON DATE
do_not_know		

Concluded