Natural Language Processing
Assignment 2

Michelle Falzon

525296M

Table of Contents

1. Introduction

Communication between components in a computer system is done mainly in binary and other low level languages. This hinders humans from communicating with electronic devices easily and quickly. Natural language processing is aimed towards bridging this gap between people and devices by programming computers to work with natural languages such as English. The program described in the later sections allows the user to ask questions about arrival or departure flights in English, parses the input, queries the database and returns answers accordingly.

2. Tasks

2.1 Building Prolog Database

The old Airport website was parsed using the SWI Prolog inbuilt functions – http_load_html() and scrape_no_error. The first function returned the DOM structure of the website. This DOM was in the form of a list with each of its elements having the following structure:

element(type, attributes, content).

The type referred to the tag of the element such as div or script, the attributes held the class, id, and style information and the content held either another child element or actual text. This DOM was parsed until the table of arrivals was found. Once the table was found, it was passed to process_table() to obtain the information about each flight and assert it to the database. The craft information was not used and thus was discarded.

```
parse_table(element(_,_,T)) :-
        airline(T,T1,[Airline|_]),
        place(T1,T2,[Place|_]),
        flightno(T2,T3,[Flightno|_]),
        craft(T3,T4,_),
        scheduledtime(T4,T5,[ScheduledTime|_]),
        arrivaltime(T5,T6,[ArrivalTime|_]),
        status(T6,_,[Status|_]),
assertz(arrival(Airline,Place,Flightno,ScheduledTime,ArrivalTime,Status)).
parse_table([]).
```

The departures database was built manually by using departure() predicates. Each predicate held information about a single flight corresponding to departure from Luqa Airport. The predicate was of the form:

departure(airline, destination, flight number, arrival time, departure time, current status).

2.2 Development of DCG Grammar

The program deals with two types of questions. Q1 refers to yes/no questions and Q2 deals with open-ended questions.

2.2.1 Yes/No Questions

Q1 consists of an auxiliary (does,did,do), a noun phrase and a verb phrase. For example: Did flights arrive from 'ZURICH'?

```
s --> aux,np,vp.
```

Noun phrases consist of either a determinant (a,the) and a noun or a determinant, a company (ex: 'AIR MALTA') and noun or a flight and its number or simply a noun on its own.

```
np --> det, n.                  ex: the flight
np --> f.                       ex: flight 'FR 112'
np --> n.                       ex: 'ROME'
np --> det, comp, n.            ex: the 'AIR MALTA' flight
```

A verb phrase is composed of a verb or a verb and a prepositional phrase or vice versa.

```
vp --> v.                       ex: arrive
vp --> v, pp.                   ex: arrived from
vp --> pp, v.                   ex:  from 'ZURICH' arrive
```

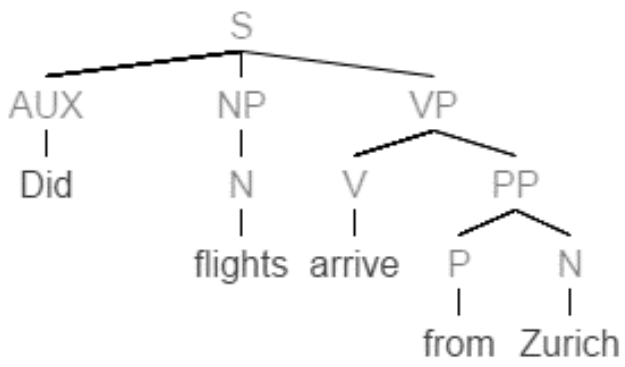
A prepositional phrase is made up of a preposition and noun.

```
pp --> p,n.                     ex: from 'ROME'
```

By combining auxiliaries, noun phrases, verb phrases and prepositional phrases, yes/no questions are formed.

Example: Did flights arrive from 'ZURICH'?

Semantic Tree:



2.2.2 Open-ended Questions

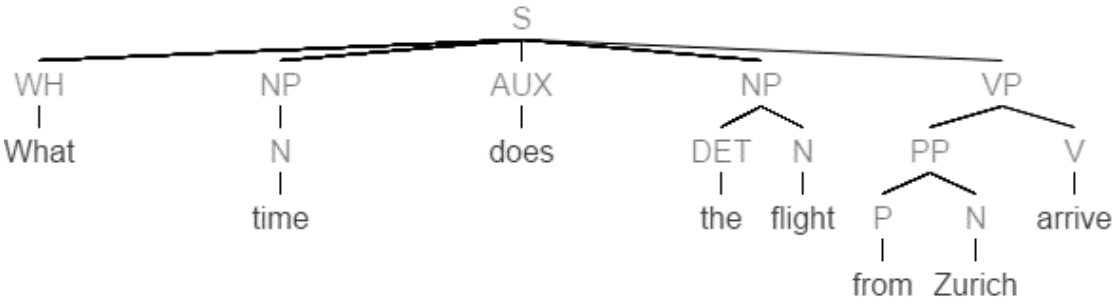
Q2 deals with questions starting with 'what', 'when' and 'how many'.

```
S --> wh, aux, np, vp.          ex: when did flight 'FR 23' arrive?
S --> wh, np, vp.               ex: what time does the flight from
                                'ZURICH' arrive?
s --> [how,many],np,vp.         ex: how many flights are there from
                                'ROME'?
```

Accepted question words starting with 'wh' are defined as follows:

```
wh --> [when].
wh --> [what, time].
wh --> [what].
```

As before, noun phrases, auxiliaries, verb phrases and prepositional phrases are used to represent a question.



2.3 Adding Semantics to Grammar

A grammar requires semantics in order to ensure that the correct tense is identified and the right function call is made. This is done by passing 'present' or 'past' according to the auxiliary used. For example if the question starts with the word 'did', the aux() predicate returns the variable T as 'past'. This is then passed to the verb phrase predicate in order to select the correct function since 'arrive' indicates both the past and present tense thus solving the issue of ambiguity.

Furthermore, the semantics is used to identify the type of noun passed. In this system the noun may refer to a place, an airline company or a flight (ex: flight 'FR 124'). Therefore, the variable 'I' returned from the noun phrase is used to store information about the noun passed. It is also used to distinguish between a yes/no question and a

question contains a place and requires the time to be returned. (Ex: What time does the flight from 'ZURICH' arrive?)

```
s(q1(S)) --> aux(T),np(NP,I,_),vp(T,I,VP), {reduce(VP,NP,S)}.
```

The reduce() function is used to apply the noun to the verb phrase and thus pass the correct arguments to the function to query the database. For example if the verb phrase consisted of X ^ P ^ arrived(X,P) and the noun was 'ZURICH', the reduced result will be X ^ arrive(X,'ZURICH'). The 'arrive' function is then called to query the database.

For 'wh' – questions 'ptime' is passed to the show that a place has been given while 'qtime' denotes that a flight number is has been passed. Both require the time of the flight to be returned. 'flightno' is used to denote that a place has been passed and that the user expects a list of flight numbers denoting flights going to that place to be returned. For 'how many' questions, 'count' is passed to indicate that a number should be returned.

2.4 Building DB Query System

Once the correct function has been chosen based on the tense of the verb and type of noun, the function queries the database.The following type of functions have been used for this project (showing only one tense):

1. ```
   arrive(_,P) :- arrival(_,P,_,_,_,'DELAYED').
   depart(_,P) :- departure(_,P,_,_,_,delayed).
   ```

   These functions are called when the user asks whether a flight from/to a certain place has arrived/departed respectively. 'P' holds the place. Pattern matching is then used to find clauses that match the ones on the right hand side.

2. ```
   arriveflightno(P,_,_,F) :- findall(F,arrival(_,P,F,_,_,' '),L1),
   findall(F,arrival(_,P,F,_,_,'DELAYED'),L2), append(L1,L2,F).

   departedflightno(P,F,_):-
   findall(F,departure(_,P,F,_,_,departed),L1),
   findall(F,departed(_,P,F,_,_,airborne),L2), append(L1,L2,F).
   ```

   These are used when the user inquires which flights has left/arrived from/to a place. 'P' holds the place and F will hold the flight number to be returned. The system uses the inbuilt findall() method in order to concatenate all instances in the database that match the second argument. The two lists; one for delayed and one for airborne (written as ' ') in the case of arrivals and one for departed and airborne for departures, are merged using the append() to give the final list of flight numbers.

3. ```
   arriveflight(_,F) :- arrival(_,_,F,_,_,'DELAYED').
   departflight(_,F) :- departure(_,_,F,_,_,airborn).
   ```
   ```

The above predicates are called when the user provides a flight number and would like to know if they have left/arrived. Once again pattern matching is used to find the clauses that contain the right flight number. Yes/no is returned accordingly.

4.
```
arrivetime(_,F,T) :- arrival(_,_,F,_,T,'DELAYED').
departtime(T,F) :- departure(_,_,F,_,T,airborne).
```

These functions accept a flight number (F) as an argument and return the time of arrival/departure of that flight by searching the database.

5.
```
arrivetimeplace(P,_,_,T) :- arrival(_,P,_,_,T,' ').
departedtimeplace(P,T,_) :- departure(_,P,_,_,T,landed).
```

Arrivedtimeplace' and 'departedtimeplace' are called when the user asks for the time of arrival/departure of a flight leaving from a given place 'P'.

6.
```
arrivecomp(C,P) :- arrival(C,P,_,_,_,'DELAYED').
departcomp(C,P,T) :- departure(C,P,_,_,T,delayed).
```

This function is called when the user provides and company and a place. It is used to check whether that flight has arrived/departed. Yes/no is returned according to whether a match has been found.

7.
```
arrivecount(P,_,_,Sum) :- findall(P,arrival(_,P,_,_,_,_), List),
listlen(List,Sum).
```

This last function computes the number of flights arriving from a place by searching the database for any flights with the arrival destination set to place 'P' and then counting the number of elements in the returned list.

2.5 Running the program

The user can run the program by using the process predicate with the question and unbound variable as arguments. Since the arrivals database is built automatically, places, airline companies and flight numbers must be written in quotation marks and must be capitalised. Those for departures are to be written in lowercase and no quotation marks are to be used.

The question should be written in list form and the unbound predicate will be bound to the answer if the query is successful or to 'no' if no matching entries were found.

process([did,flights,leave,for,rome], Answer).

The process() predicate in turn calls s() and then interpret(). The former is used to parse the question provided by the user and returns the query as a result which is then interpreted by querying the database.

3. Results

The system works as expected. The following are some of the valid questions accepted by the system:

- Are flights arriving from ‘ZURICH’?
- Are flights leaving from Zurich?

- Do flights arrive from ‘ZURICH’?
- Do flights leave from Zurich?

- Does flights arrive from ‘ZURICH’?
- Does flights leave from Zurich?

- Did flights arrive from ‘ZURICH’?
- Did flights leave from Zurich?

- When does flight ‘KM 481’ arrive?
- When does flight KM481 leave?

- Did the ‘AIR MALTA’ flight from ‘ZURICH’arrive?
- Did the Airmalta flight from Zurich leave?

- What time does the flight from ‘ZURICH’arrive?
- What time does the flight from Zurich leave?

- What flights arrive from ‘ZURICH’?
- What flights leave from Zurich?

- What flights arrived from ‘ZURICH’?
- What flights left from Zurich?

- How many flights are there from ‘BRUSSELS’?

4. Conclusion and Future Work

The system, though functional, may be further improved on by also obtaining the departure flight information in real time from the online website. It may also be made more robust by dealing with more variations of the same question as well as handling incorrect inputs gracefully. Lastly, the system would be more intuitive and easy to use if the replies are written in sentence form.

5. References

Image: http://www.rubenshotel.com/~/media/ttc/rch/the%20milestone/main%20carousel/desktop/airport_1400x568.jpg?h=568&la=en-US&w=1400

Obtaining HTML: http://www.swi-prolog.org/howto/scrape/ParseHTML.html

Findall(): http://www.swi-prolog.org/pldoc/doc_for?object=findall/3