Natural Language Processing Term Project (Part 2)

Team members:
Amruta Nanavaty(UIN: 665473328**)**
Sabita Acharya(UIN: 676636765**)**

PROCESS:

The first task was to identify the type of question that was being asked by the user. If we observe the parse tree and the dependency structures provided by Stanford for each sentence, we can see that there is certain repeating pattern in each category of sentences. For example: the dependency structure obtained for following questions are:

"Who won gold(Y) in speedskating(Z)?"

nsubj(won-2, Who-1)
root(ROOT-0, won-2)
dobj(won-2, gold-3)
prep(won-2, in-4)
pobj(in-4, speedskating-5)

"Who arrived first in speedskating?"

nsubj(arrived-2, Who-1)
root(ROOT-0, arrived-2)
advmod(arrived-2, first-3)
prep(arrived-2, in-4)
pobj(in-4, speedskating-5)

The parse trees that we obtain for these examples are also similar. A Wh-question will have a "WP" at the beginning of its parse tree and a Yes/No question with complex subjects will have a determiner (DET) followed by an adjective (JJ) due to its components like "a Canadian man", "a Russian woman" etc. If a question does not fall under any one of the above categories and contains an "Aux" in the dependency structure, we can identify it as a Yes/No question with only proper or bare nouns. Also, from all the examples that we tried out, we found out that if verbs like "win" or "won" arrive in the question, then the dependency structure will have a "dobj" with win/won as the governor(first component) and the position/type of medal won as the dependent(second) component. Similarly, arrive/arrived occurs as the governor in "advmod" component of the dependency structure.

If we make use of these similarities that occur in sentences of a particular type, then the task of developing semantic attachments is reduced to identifying the constituting semantic components of the sentence and extracting the reasonable values from dependency structure. The main difficulty that we faced was due to the random behavior of words that are expected to fall under a similar category. For example, words like first, second, etc. occur within different components of the dependency structure. So we need to carefully analyze all the cases and create a general method that addresses all of them. Also, we need to map nationalities to the nation name i.e. "Canadian" should be mapped to "Canada". For this task, the "like" operator of SQL query could not be used because there are certain nationality-nation pairs like "American" and "USA" where the two words are not similar to each other in any aspect. So we extracted all the nation names present in the database, found their nationalities and saved it in a text file for reference.

Once all the necessary components were extracted from the dependency structure, we plugged in those values into SQL statements and searched into the database for an answer. If the question asked is not according to the format or no result is obtained from the database, the user is informed and is asked to enter another question.

SAMPLE OUTPUT:

Query: Did Mancuso arrive third in super-combined?
<SQL> : Select count(*) from results R INNER JOIN competitions C on R.comp_id=C.comp_id where R.winner like '%mancuso%' and R.medal= 'bronze' and C.name= 'super-combined';
<ANSWER> : yes

Query: Did Katsalapov win silver in icedancing?
<SQL> : Select count(*) from results R INNER JOIN competitions C on R.comp_id=C.comp_id where R.winner like '%katsalapov%' and R.medal= 'silver' and C.name= 'icedancing';
<ANSWER> : yes

Query: Who arrived third in super-combined?
<SQL> : Select R.winner from results R INNER JOIN competitions C on R.comp_id=C.comp_id where R.medal= 'bronze' and C.name= 'super-combined';
<ANSWER> : mancuso

Query: Who won silver in giantslalom?
<SQL> : Select R.winner from results R INNER JOIN competitions C on R.comp_id=C.comp_id where R.medal= 'silver' and C.name= 'giantslalom';
<ANSWER> : missillier

Query: Did a Canadian woman win silver in icedancing?
<SQL> : Select count(*) from athletes A INNER JOIN results R ON A.name=R.winner INNER JOIN competitions C on R.comp_id=C.comp_id where R.medal= 'silver' and C.name= 'icedancing'and A.nationality= 'canada' and A.gender= 'F';
<ANSWER> : no

Query: Did a Chinese man arrive second in shorttrack?
<SQL> : Select count(*) from athletes A INNER JOIN results R ON A.name=R.winner INNER JOIN competitions C on R.comp_id=C.comp_id where R.medal= 'silver' and C.name= 'shorttrack'and A.nationality= 'china' and A.gender= 'M';
<ANSWER> : yes

EXTRA CREDIT

The program can handle all the categories of questions (Wh-questions, Yes/No questions with only proper and bare nouns and Yes/No questions with complex subjects) pertaining to the formats mentioned below:

- Who won gold in 500m speedskating?
- Who arrived first in 500 speedskating?
- Did a Chinese man win silver in nearhill skijumping?
- Did a Russian woman arrive third in lh skijumping?
- Did Semerenko win bronze in 7500m biathlon?
- Did Mattel arrive third in largehill skijumping?