Maciej Girek - Mgirek2
Manuel Adrianzen - Madria3
CS - 421
Partner Project Part 2

Approach

By using the Stanford parser to get the Part of Speech (POS) tags for the words we were able to differentiate which words are nouns, proper nouns and wh-pronouns etc. With the tags associated to the words we created a function called properNounNER that tell us more details about the proper nouns by utilizing the Stanford named entity recognizer (NER) to determine if the proper noun is a person name, movie, or date. However, stanford parser alone did not help make the relationship between a preposition and a noun such as ‘which actress’ or ‘in 2007’. For us to be able to join this kind of syntactic relationship we needed to use a Regex Parser with a pattern that is able to recognize POS and join words together to form a phrase such as the prepositional phrase “in 2007”. The regex parser works by producing a pattern of tags with regular expression to fit the phrase description. Regular Expressions are a particular kind of formal grammar used to parse strings and other textual information that are known as "Regular Languages" in formal language theory. Once all the words were grouped and tagged we created a set of booleans that get set to true when the booleans correspond to the specific sentence type. Depending on which booleans are set to true we send the keywords of the sentence to a function that performs the corresponding SQL query using those keywords.

Example of the Pattern to form Phrases

```
pattern = """NP: {(<DT>?<NNP>+<JJR>) | (<DT>?<NN>+<JJR>)
    VBD: {<VBD>}
    PP: { <IN><NP> | <IN>}
    IN: {<IN>}
    VP: {<V> | <NP> | <PP> | <NP><PP>+ }"""
```

Code to create the phrase tree

```
    print("<PARSETREE>")
    result = nltk.pos_tag(nltk.word_tokenize(s))
    NPChunker = nltk.RegexpParser(pattern)
    result = NPChunker.parse(result)
    Tree.fromstring(str(result)).pretty_print()
```

Issues

One of the issues we discovered is that the method of sending queries through python as a string required us to adjust the sentences a little such as inserting a backslash (\) before an apostrophe so that the string differentiates the apostrophe as part of the word and not a SQL indicator for the name of something.

EX) Schindler's list → Schindler\'s list

This is due to the way SQL uses the WHERE statement and a single quote to determine the title of something.

EX) WHERE Person.name LIKE '%Kubrick'

We also discovered a third party SQL view was manipulating the data somehow and caused us to have issues between the SQL databases. We resolved this by re-downloading the database and replacing it in the git so we have the same database to work with. Ultimately, we found the SQL Lite manager addon for Mozzilla Firefox recommended by the professor to be very helpful in helping us experiment with queries to answer the questions.

The following natural language queries are the queries before we send the appropriate keyword to the function to insert it and perform the query. Numbering corresponds to the Results section for questions.

Natural Language Query YES / NO Questions

1. ```
   "Select Count(*) FROM Person P INNER JOIN Director D ON P.id = director_id INNER JOIN Movie M ON M.id = movie_id WHERE P.name like '%"+person+"%'"
   ```
2. ```
   "Select Count(*) FROM Movie M INNER JOIN Director D ON D.movie_id = M.id INNER JOIN Person P ON P.id = D.director_id WHERE P.name LIKE '%"+director+"%' AND M.name LIKE '%"+movie+"%'"
   ```
3. ```
   "Select Count(*) FROM Movie M INNER JOIN Actor A ON A.movie_id = M.id INNER JOIN Person P ON P.id = A.actor_id WHERE P.name LIKE '%"+name+"%' AND M.name LIKE '%"+movie+"%'"
   ```
4. ```
   "Select Count(*) FROM Oscar O INNER JOIN Person P ON person_id = O.person_id Where P.name like '%"+name+"%' and O.year = '"+year+"'"
   ```
5. ```
   "Select Count(*) FROM Actor A INNER JOIN Oscar O ON O.person_id = A.actor_id INNER JOIN Person P ON P.id = A.actor_id WHERE O.year LIKE '%"+year+"%' AND P.pob LIKE '%"+country+"%'"
   ```
6. ```
   """select count(*) from oscar
   inner join movie on movie.id = oscar.movie_id
   inner join actor on actor.movie_id = movie.id
   inner join person on person.id = actor.actor_id where oscar.type = '"""+oscar+"""' and person.name like '%"""+name+"""%'"""
   ```

Natural Language Queries WH- Questions

1. """select person.name from person
   inner join director on person.id = director.director_id
   inner join movie on movie.id = director.movie_id
   where movie.name like '%"""+movie+"""%'"""
2. """select person.name from director
   inner join person on person.id = director.director_id
   inner join oscar on oscar.person_id = person.id
   inner join movie on movie.id = oscar.movie_id
   where oscar.movie_id In (select movie_id from oscar
   where oscar.type like '"""+oscar+"""' and
       oscar.year = """+year+""")"""
3. "select person.name from person
inner join oscar on oscar.person_id = person.id
inner join movie on movie.id = oscar.movie_id
where oscar.year = "+year+" And oscar.type like '"+award+"'"
4. """select person.name from person
   inner join oscar on oscar.person_id = person.id
   inner join movie on movie.id = oscar.movie_id
   where oscar.movie_id In (select movie_id from oscar
   where oscar.year = """+year+""" and oscar.type like
'"""+oscar+"""')"""
5. """select movie.name from oscar
   inner join movie on movie.id = oscar.movie_id
   where oscar.movie_id In (select movie_id from oscar
   where oscar.year = """+year+""") And oscar.type =
'"""+oscar+"""'"""
6. """select oscar.year from oscar
   inner join movie on movie.id = oscar.movie_id
   inner join person on person.id = oscar.person_id
   where oscar.movie_id In (select movie_id from oscar
   where oscar.type like '"""+oscar+"""')and
       person.name like '%"""+name+"""%'"""

Results for YES / NO Questions and Constructed query

1) **Is Kubrick a director?**

```
<QUESTION> Is Kubrick a director?
<QUERRY>
Select Count(*) FROM Person P INNER JOIN Director D ON P.id = director_id INNER JOIN Movie M ON M.id = movie_id WHERE P.name like '%Kubrick%'
<ANSWER> Yes
```

2) **Is Mighty Aphrodite by Allen?**

```
<QUESTION> Is Mighty Aphrodite by Allen?
<QUERRY>
Select Count(*) FROM Movie M INNER JOIN  Director D ON D.movie_id = M.id INNER JOIN  Person P ON P.id = D.director_id WHERE P.name LIKE '%Allen%' AND
<ANSWER> Yes
```

3) **Was Loren born in Italy?**

```
<QUESTION> Was Loren born in Italy?
<QUERRY>
Select Count(*) FROM Person P WHERE P.name LIKE '%Loren%' AND P.pob LIKE '%Italy%'
<ANSWER> Yes
```

4) **Was Birdman the best movie in 2015?**

```
<QUESTION> Was Birdman the best movie in 2015?
<QUERRY>
Select Count(*) FROM Oscar O INNER JOIN  Movie M ON movie_id = M.id Where M.name like '%Birdman%' and O.type = 'BEST-PICTURE'
<ANSWER> Yes
```

5) **Did Neeson star in Schindler\'s List?**

```
<QUESTION> Did Neeson star in Schindler\'s List?
<QUERRY>
Select Count(*) FROM Movie M INNER JOIN  Actor A ON A.movie_id = M.id INNER JOIN  Person P ON P.id = A.actor_id WHERE P.name LIKE '%Neeson%' AND M.nam
<ANSWER> Yes
```

6) **Did Swank win the oscar in 2000?**

```
<QUESTION> Did Swank win the oscar in 2000?
<QUERRY>
Select Count(*) FROM Oscar O INNER JOIN  Person P ON person_id = O.person_id Where P.name like '%Swank%' and O.year = '2000'
<ANSWER> Yes
```

7) **Did a French actor win the oscar in 2012?**

```
<QUESTION> Did a French actor win the oscar in 2012?
<QUERRY>
Select Count(*) FROM Actor A INNER JOIN Oscar O ON O.person_id = A.actor_id INNER JOIN  Person P ON P.id = A.actor_id WHERE O.year LIKE '%2012%' AND P.pob LIKE '%France%'
<ANSWER> Yes
```

8) **Did a movie with Neeson win the oscar for best film?**

```
<QUESTION> Did a movie with Neeson win the oscar for best film?
<QUERRY>
select count(*)  from oscar
       inner join movie on movie.id = oscar.movie_id
       inner join actor on actor.movie_id = movie.id
       inner join person on person.id = actor.actor_id where oscar.type = 'BEST-PICTURE' and person.name like '%Neeson%'
<ANSWER> Yes
```

Results for WH-Questions

1. **Who directed Schindler's List?**

```
<QUESTION> Who directed Schindler\'s List?
<QUERRY>
select person.name from person
       inner join director on person.id = director.director_id
       inner join movie on movie.id = director.movie_id
       where movie.name like '%List%'
<ANSWER> Steven Spielberg
```

2. **Who won the oscar for best actor in 2005?**

```
<QUESTION> Who won the oscar for best actor in 2005?
<QUERRY>
select person.name from person inner join oscar on oscar.person_id = person.id inner join movie on movie.id = oscar.movie_id where oscar.year = 2005 And oscar.type like 'BEST-ACTOR'
<ANSWER> Jamie Foxx
```

3. **Who directed the best movie in 2010?**

```
<QUESTION> Who directed the best movie in 2010?
<QUERRY>
select person.name from director
       inner join person on person.id = director.director_id
       inner join oscar on oscar.person_id = person.id
       inner join movie on movie.id = oscar.movie_id
       where oscar.movie_id In (select movie_id from oscar
       where oscar.type like 'BEST-PICTURE' and
           oscar.year = 2010)
<ANSWER> Kathryn Bigelow
```

4. **Which actress won the oscar in 2012?**

```
<QUESTION> Which actress won the oscar in 2012?
<QUERRY>
select person.name from person
       inner join oscar on oscar.person_id = person.id
       inner join movie on movie.id = oscar.movie_id
       where oscar.movie_id In (select movie_id from oscar
       where oscar.year = 2012 and oscar.type like 'BEST-ACTRESS')
<ANSWER> Meryl Streep
```

5. **Which movie won the oscar in 2000?**

```
<QUESTION> Which movie won the oscar in 2000?
<QUERRY>
select movie.name from oscar
       inner join movie on movie.id = oscar.movie_id
       where oscar.movie_id In (select movie_id from oscar
       where oscar.year = 2000) And oscar.type = 'BEST-PICTURE'
<ANSWER> American Beauty
```

6. **When did Blanchett win an oscar for best actress?**

```
<QUESTION> When did Blanchett win an oscar for best actress?
<QUERRY>
select oscar.year from oscar
       inner join movie on movie.id = oscar.movie_id
       inner join person on person.id = oscar.person_id
       where oscar.movie_id In (select movie_id from oscar
       where oscar.type like 'BEST-ACTRESS')and
           person.name like '%Blanchett%'
<ANSWER> 2014
```