CS421
University of Illinois at Chicago
Spring 2019
Omar Al-Khatib / oalkha2
Isaiah Grief / igrief2
Michael Hume / mhume3

Report

We used the CoreNLP parser interfacing with NLTK in Python. We used CoreNLP because it seemed that the CoreNLP parser was robust and parsed sentences in a way that made displaying the parse trees easy using NLTK.

The CoreNLP parser was useful for tasks such as Part Of Speech tagging and parsing sentences. NLTK was used to convert the parsed list of words output by CoreNLP and turn the list into a tree. The tree could then be traversed using NLTK tree attributes.

For the categorization task, WordNet was used to determine which of three categories a sentence was most similar to. WordNet was used because it provided an easy way to use the Wu-Palmer algorithm to calculate similarity between words. Similarity between words in input sentences and words in premade lists was averaged and the sentences were placed into the category with the highest average similarity. A limited number of words from each list were used to calculate the similarity, in order to not bias the results depending on the length of the premade lists. In order to ensure that the questions for part 2 of the project could be answered, some additional category filtering was added based on keywords after the first categorization process.

Construction of queries was modular and based on the type of question asked. The WordNet lemmatizer was used to determine the lemma of a verb at the beginning of the sentence. A ‘do’ or ‘be’ lemma indicated a yes or no question, and all other questions for the sake of the assignment were assumed to be WH- questions. For yes or no questions the query always began with SELECT COUNT(*) FROM, and WH- questions varied depending on the specific word. ‘who’ would select for a person, so depending on the category, we would look for the table that contained the name of a person and selected that.

A few FROM statements for queries were created and included as many joins as possible. For Movies, a from statement for Movie was created, as well as a statement joining Movie and Oscar. Two additional functions could join Director to Movie or Actor to Movie. In order to determine if the sentence indicated that the Oscar table was needed, the sentences were examined for adjectives like ‘best’, as well as nationalities like ‘French’, ‘American’, and ‘Italian’. Using the word ‘best’ we examined NP -> JJS NN relationships to figure out what Oscar category the sentence might be talking about. These categories were also useful in figuring out if the sentence’s subject was a person, film, or director.

We examined NP -> CD sections of the parse tree and used information previously gathered on whether or not the sentence discussed an Oscar in order to determine what year to query for. These were WHERE clauses in the queries. Another where clause was the nationalities that were found in a previous section. These adjectives denoted a specific origin country. For Movies, place of birth was queried for using the origin country obtained through the adjective.

Part of speech tags were used along with keywords like specific verbs that required specific inputs of arguments. These verbs usually allowed for proper nouns to be inserted as arguments into queries. A process of combining proper nouns and discarding possessives was done to the sentences in order to get rid of any issues that multi-word proper nouns had, particularly for the cases when the parser had trouble figuring out how to build the tree with these noun phrases in mind. Combining the words into a single noun phrase allowed these phrases to be substituted directly into the queries. This implementation had problems with proper nouns mixing with regular nouns. The CoreNLP parser is not perfect, and so a lot of issues with specific noun phrases arose, and it was difficult to design queries using parse trees that were not quite correct.

Example output:

```
<QUESTION> Is Rome the capital of Italy?

<QUERY>
SELECT COUNT(*) FROM Capitals INNER JOIN Cities ON Cities.Id = Capitals.CityID
INNER JOIN Countries ON Countries.Id = Capitals.CountryID WHERE Cities.Name = 'Rome'
and Countries.Name = 'Italy'
<ANSWER>
Yes
*   *   *   *   *   *   *   *   *   *   *   *
<QUESTION> Is France in Europe?

<QUERY>
SELECT COUNT(*) FROM Countries INNER JOIN CountryContinents ON Countries.Id =
CountryContinents.CountryID INNER JOIN Continents ON CountryContinents.ContinentID =
Continents.Id WHERE Countries.Name = 'France' and Continents.Continent = 'Europe'
<ANSWER>
Yes
*   *   *   *   *   *   *   *   *   *   *   *
<QUESTION> Did Madonna sing PapaDoNotPreach?

<QUERY>
SELECT COUNT(*) FROM Artist INNER JOIN Album ON Artist.id = Album.artsitID INNER
JOIN Track ON Track.albumID = Album.albumID  WHERE Artist.name LIKE '%Madonna%'
AND Track.name LIKE '%PapaDoNotPreach%'
<ANSWER>
No
*   *   *   *   *   *   *   *   *   *   *   *
```

CS421

University of Illinois at Chicago

Spring 2019

Omar Al-Khatib / oalkha2

Isaiah Grief / igrief2

Michael Hume / mhume3

<QUESTION> Does the album Thriller include the track BeatIt?

<QUERY>

```
SELECT COUNT(*) FROM Artist INNER JOIN Album ON Artist.id = Album.artsitID INNER
JOIN Track ON Track.albumID = Album.albumID
```

<ANSWER>

Yes

*   *   *   *   *   *   *   *   *   *   *   *

<QUESTION> Was Beyonce' born in the USA?

<QUERY>

```
SELECT COUNT(*) FROM
```

<ANSWER>

I don't know

*   *   *   *   *   *   *   *   *   *   *   *

<QUESTION> What is the capital of Spain?

<QUERY>

```
SELECT Cities.Name FROM Capitals INNER JOIN Cities ON Cities.Id = Capitals.CityID
INNER JOIN Countries ON Countries.Id = Capitals.CountryID WHERE Countries.Name =
'Spain'
```

<ANSWER>

I don't know

*   *   *   *   *   *   *   *   *   *   *   *

<QUESTION> Where is Rome?

<QUERY>

```
SELECT Countries.Name FROM Capitals INNER JOIN Cities ON Cities.Id = Capitals.CityID
INNER JOIN Countries ON Countries.Id = Capitals.CountryID WHERE Cities.Name = 'Rome'
```

<ANSWER>

Italy

*   *   *   *   *   *   *   *   *   *   *   *

<QUESTION> Is Kubrick a director?

<QUERY>

```
SELECT COUNT(*) FROM Artist INNER JOIN Album ON Artist.id = Album.artsitID INNER
JOIN Track ON Track.albumID = Album.albumID
```

<ANSWER>

Yes

CS421
University of Illinois at Chicago
Spring 2019
Omar Al-Khatib / oalkha2
Isaiah Grief / igrief2
Michael Hume / mhume3

* * * * * * * * * * * *
<QUESTION> Is Mighty Aphrodite by Allen?

<QUERY>
```
SELECT COUNT(*) FROM Movie INNER JOIN Actor ON Movie.id = Actor.movie_id INNER
JOIN Person ON Actor.actor_id = Person.id  WHERE Person.name LIKE '%Mighty
Aphrodite%' AND Movie.name LIKE '%Allen%'
```
<ANSWER>
No
* * * * * * * * * * * *
<QUESTION> Was Loren born in Italy?

<QUERY>
```
SELECT COUNT(*) FROM
```
<ANSWER>
I don't know
* * * * * * * * * * * *
<QUESTION> Was Birdman the best movie in 2015?

<QUERY>
```
SELECT COUNT(*) FROM Movie INNER JOIN Oscar ON Movie.id = Oscar.movie_id
WHERE Movie.name LIKE '%Birdman%' AND Oscar.type = 'BEST-PICTURE' AND
Oscar.year = 2015
```
<ANSWER>
Yes
* * * * * * * * * * * *
<QUESTION> Did Neeson star in Schindler's List?

<QUERY>
```
SELECT COUNT(*) FROM Artist INNER JOIN Album ON Artist.id = Album.artsitID INNER
JOIN Track ON Track.albumID = Album.albumID  WHERE Artist.name LIKE '%Neeson%'
AND Track.name LIKE '%Schindler%'
```
<ANSWER>
No
* * * * * * * * * * * *
<QUESTION> Did Swank win the oscar in 2000?

<QUERY>

```
SELECT COUNT(*) FROM Movie INNER JOIN Oscar ON Movie.id = Oscar.movie_id
WHERE Movie.name LIKE '%Swank%' AND Oscar.year = 2000
```

<ANSWER>
No

* * * * * * * * * * * *

<QUESTION> Did a French actor win the oscar in 2012?

<QUERY>

```
SELECT COUNT(*) FROM Movie INNER JOIN Oscar ON Movie.id = Oscar.movie_id
INNER JOIN Actor ON Movie.id = Actor.movie_id INNER JOIN Person ON Actor.actor_id =
Person.id  WHERE Oscar.year = 2012 AND Person.pob LIKE '%France%'
```

<ANSWER>
Yes

* * * * * * * * * * * *

<QUESTION> Did a movie with Neeson win the oscar for best film?

<QUERY>

```
SELECT COUNT(*) FROM Movie INNER JOIN Oscar ON Movie.id = Oscar.movie_id
WHERE Movie.name LIKE '%Neeson%' AND Oscar.type = 'BEST-PICTURE'
```

<ANSWER>
No

* * * * * * * * * * * *

<QUESTION> Who directed Schindler's List?

<QUERY>

```
SELECT Person.name FROM Movie INNER JOIN Director ON Movie.id = Director.movie_id
INNER JOIN Person ON Director.director_id = Person.id  WHERE Movie.name LIKE
'%Schindler%'
```

<ANSWER>
Steven Spielberg

* * * * * * * * * * * *

<QUESTION> Who won the oscar for best actor in 2005?

<QUERY>

```
SELECT Person.name FROM Movie INNER JOIN Oscar ON Movie.id = Oscar.movie_id
INNER JOIN Actor ON Movie.id = Actor.movie_id INNER JOIN Person ON Actor.actor_id =
Person.id  WHERE Oscar.year = 2005 AND Oscar.type = 'BEST-ACTOR'
```

<ANSWER>
Jamie Foxx

CS421
University of Illinois at Chicago
Spring 2019
Omar Al-Khatib / oalkha2
Isaiah Grief / igrief2
Michael Hume / mhume3

* * * * * * * * * * * *

<QUESTION> Who directed the best movie in 2010?

<QUERY>

```
SELECT Person.name FROM Movie INNER JOIN Oscar ON Movie.id = Oscar.movie_id
INNER JOIN Director ON Movie.id = Director.movie_id INNER JOIN Person ON
Director.director_id = Person.id  WHERE Oscar.year = 2010 AND Oscar.type = 'BEST-PICTURE'
```

<ANSWER>
Kathryn Bigelow

* * * * * * * * * * * *

<QUESTION> Which American actress won the oscar in 2012?

<QUERY>

```
SELECT Person.name FROM Movie INNER JOIN Oscar ON Movie.id = Oscar.movie_id
INNER JOIN Actor ON Movie.id = Actor.movie_id INNER JOIN Person ON Actor.actor_id =
Person.id  WHERE Oscar.year = 2012 AND Person.pob LIKE '%USA%'
```

<ANSWER>
Emma Stone

* * * * * * * * * * * *

<QUESTION> Which movie won the oscar in 2000?

<QUERY>

```
SELECT Movie.name FROM Movie INNER JOIN Oscar ON Movie.id = Oscar.movie_id
WHERE Oscar.year = 2000
```

<ANSWER>
American Beauty

* * * * * * * * * * * *

<QUESTION> When did Blanchett win an oscar for best actress?
<QUERY>

```
SELECT Oscar.year FROM Movie INNER JOIN Oscar ON Movie.id = Oscar.movie_id INNER
JOIN Actor ON Movie.id = Actor.movie_id INNER JOIN Person ON Actor.actor_id = Person.id
WHERE Oscar.type = 'BEST-ACTRESS'
```

<ANSWER>
1940