Task 1: Question Answering Systems

Given are the following five queries:

- a: What movies has Leonardo DiCaprio acted in?
- b: Give me the movies whose music composers have won the BAFTA Award for Best Film Music.
- c: Who wears the number 5 of Germany?
- d: Has Germany won the FIFA World Cup 2014?
- e: Who is the US president?

Task 1.1: Describe what kind of different result types you expect for each query.

- a, b: result list
- c,e: single result
- d: yes/no (ask query)

Task 1.2: Name potential challenges for each query (if there are any).

- a: -
- b: Complex information need.
- c: Domain knowledge needed, but domain (football?) is not provided. The query result depends on the query time.
- d: Boolean queries are not very fault-tolerant, the exact triple is needed.
- e: The query result depends on the query time.

Task 1.3: Execute the queries on all of the following Question Answering systems and describe the results.

- Wolfram Alpha: http://www.wolframalpha.com
- Google: www.google.com

<table>
<thead>
<tr>
<th>Query</th>
<th>WDAqua</th>
<th>Wolfram Alpha</th>
<th>START</th>
<th>Google</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>movie list ✔️</td>
<td>movie list ✔️</td>
<td>movie list ✔️</td>
<td>movie list ✔️</td>
</tr>
<tr>
<td>b</td>
<td>no results ✗️</td>
<td>no results ✗️</td>
<td>no results ✗️</td>
<td>no results ✗️</td>
</tr>
<tr>
<td>c</td>
<td>United Kingdom ✗️</td>
<td>“a” ✗️</td>
<td>no results ✗️</td>
<td>Thomas Müller ✔️</td>
</tr>
<tr>
<td>d</td>
<td>2006 FIFA World Cup, 2014 FIFA World Cup ✗️</td>
<td>“no exchange data available” ✗️</td>
<td>no results ✗️</td>
<td>match list ( ✗️)</td>
</tr>
<tr>
<td>e</td>
<td>Donald Trump ✔️</td>
<td>Donald Trump ✔️</td>
<td>Donald Trump ✔️</td>
<td>list of US presidents ( ✗️)</td>
</tr>
</tbody>
</table>

1 the 5th letter of “Germany”
**Task 2: Semantic Question Answering Example System**

**Task 2.1:** Query for “What movies has Leonardo DiCaprio acted in?” on [wdqua-frontend.univ-st-etienne.fr](http://wdqua-frontend.univ-st-etienne.fr) and list three result.

Titanic, The Great Gatsby, Inception

**Task 2.2:** Use the “Did you mean” button to list potential interpretations of the entities included in the query.

- film: sequence of images ...
- Leonardo DiCaprio: American actor and film producer
- acting: impersonation of a fictional character
- Leonardo da Vinci: Italian Renaissance polymath
- Act: Act (2013 South Korean film)
- ...

**Task 2.3:** Use the “Q” button to find Wikidata SPARQL queries generated for this natural language query. Find a SPARQL query that gives correct results and one query that gives wrong results and confirm this using the Wikidata SPARQL endpoint ([https://query.wikidata.org/](https://query.wikidata.org/)).

**Correct query:**
```
SELECT DISTINCT ?s1 where {
} limit 1000
```

**Wrong query:**
```
SELECT DISTINCT ?o1 where {
} limit 1000
```

**Task 2.4:** What is the role of the relation linking in this example?

Relation linking is not required. For example, the first query here does not have constraints on the predicate. Therefore, the film “Runner, Runner” is included in the results list, although DiCaprio has not acted in it, but produced it.
**Task 3: Question Answering Pipeline**

Given is the natural language query "Which papers has Michelle written?" and the following triple database:

Charlie Carpenter, write, Contributions of Michelle
Charlie Carpenter, write, Keyword Search in XML
Michael Richardson, write, Keyword Search in XML
Michelle, write, Keyword Search in XML
Michelle, write, Pattern Matching in XML
Michelle, write, Algorithms for TopK Query
Keyword Search in XML, cite, Contributions of Michelle
Pattern Matching in XML, cite, Contributions of Michelle
Keyword Search in XML, cite, Pattern Matching in XML
Pattern Matching in XML, cite, Algorithms for TopK Query
Keyword Search in XML, cite, Algorithms for TopK Query
Contributions of Michelle, type, Paper
Keyword Search in XML, type, Paper
Pattern Matching in XML, type, Paper
Algorithms for TopK Query, type, Paper
Charlie Carpenter, type, Person
Michael Richardson, type, Person
Michelle, type, Person
Charlie Carpenter, type, Woman
Michelle, type, Woman
Michael Richardson, type, Man

**Task 3.1** What should be the output of the entity linking step?

Michelle → <Michelle>
papers → <Paper>

**Task 3.2:** What should be the output of the relation linking step?

written → <write>

**Task 3.3** What should be the top ranked SPARQL query?

```sparql
SELECT ?a WHERE {
    <Michelle> <write> ?a .
    ?a rdf:type <Paper> .
}
```
Task 4: Template-based Benchmark Generation

For each of the following three templates:

- Create an example query based on the dataset in Task 3.
- Write down the corresponding natural language question for your query.
- Write down the expected results retrieved by your query.

- a: simple query

  ```sparql
  SELECT DISTINCT ?uri WHERE {
    <a> <b> ?x .
    ?x <c> ?uri
  }
  ```

  a: Which papers have been cited by Michelle?

  ```sparql
  SELECT DISTINCT ?uri WHERE {
    <Michelle> <write> ?x .
    ?x <cite> ?uri
  }
  ```

  → Contributions of Michelle, Pattern Matching in XML, Algorithms for TopK Query

- b: count query

  ```sparql
  SELECT DISTINCT COUNT(?uri) WHERE {
    ?uri <a> <b> .
    ?uri rdf:type <class> .
  }
  ```

  b: How many women have written the paper “Keyword Search in XML”?

  ```sparql
  SELECT DISTINCT COUNT(?a) WHERE {
    ?uri <write> <Keyword_Search_in_XML> .
    ?uri rdf:type <Woman> .
  }
  ```

  → 2

- c: ask query

  ```sparql
  ASK WHERE {
  }
  ```

  c: Has Michelle written the paper “Keyword Search in XML”?

  ```sparql
  ASK WHERE {
    <Michelle> <write> <Keyword_Search_in_XML> .
  }
  ```

  → yes

---

2 [https://github.com/AskNowQA/LC-QuAD/blob/develop/templates.py](https://github.com/AskNowQA/LC-QuAD/blob/develop/templates.py)