Project: Query Answering over Linked Data

Using linked data database to answer real world questions.

Polytech Paris Saclay
Germain Rullier - APP5 Info

Question 1

`"Which car is called a “duck” in German?"` .

To solve this problem I had to answer multiple interrogations I had, see below.

What source of data should I use ?

DBPedia

- https://www.dbpedia.org/.
- Referenced during our semantic web course
- DBPedia doesn't provide a way to manually research entities through a search bar
- Only way I found to browse existing data was through SPARQL queries.
- That appeared too hard for me

What source of data should I use ?

WikiData

- https://www.wikidata.org/.
- Offers a search bar
- We can type in the word `duck`
- Unique identifier is assiocated to the duck entity: `Q3736439` .

duck (Q3736439)

common name for many species in the bird family Anatidae

Main page
Community portal
Project chat
Create a new Item
Recent changes
Random Item
Query Service
Nearby
Help
Donate

Lexicographical data
Create a new Lexeme
Recent changes

In more languages

Configure

Language	Label	Description	Also known as
English	duck	common name for many species in the bird family Anatidae	
French	canard	terme générique qui désigne des oiseaux aquatiques	
Spanish	pato	especie de pájaro que suele habitar en lagos	
German	Enten	No description defined	Ente

How to translate my problem in a SPARQL query ?

- Identify the features of the SPARQL language that will be useful for my problem.
- `"Which car is called a “duck” in German?"` .

String comparaison

- Searched on a search engine `"comparing strings in sparql"` and found the following function:

```
FILTER(ex:ldistance(?string1, ?string2) < 2)
```

- Returns `true` if the two string provided have a `Levenshtein distance` inferior to 2.

String comparaison (2)

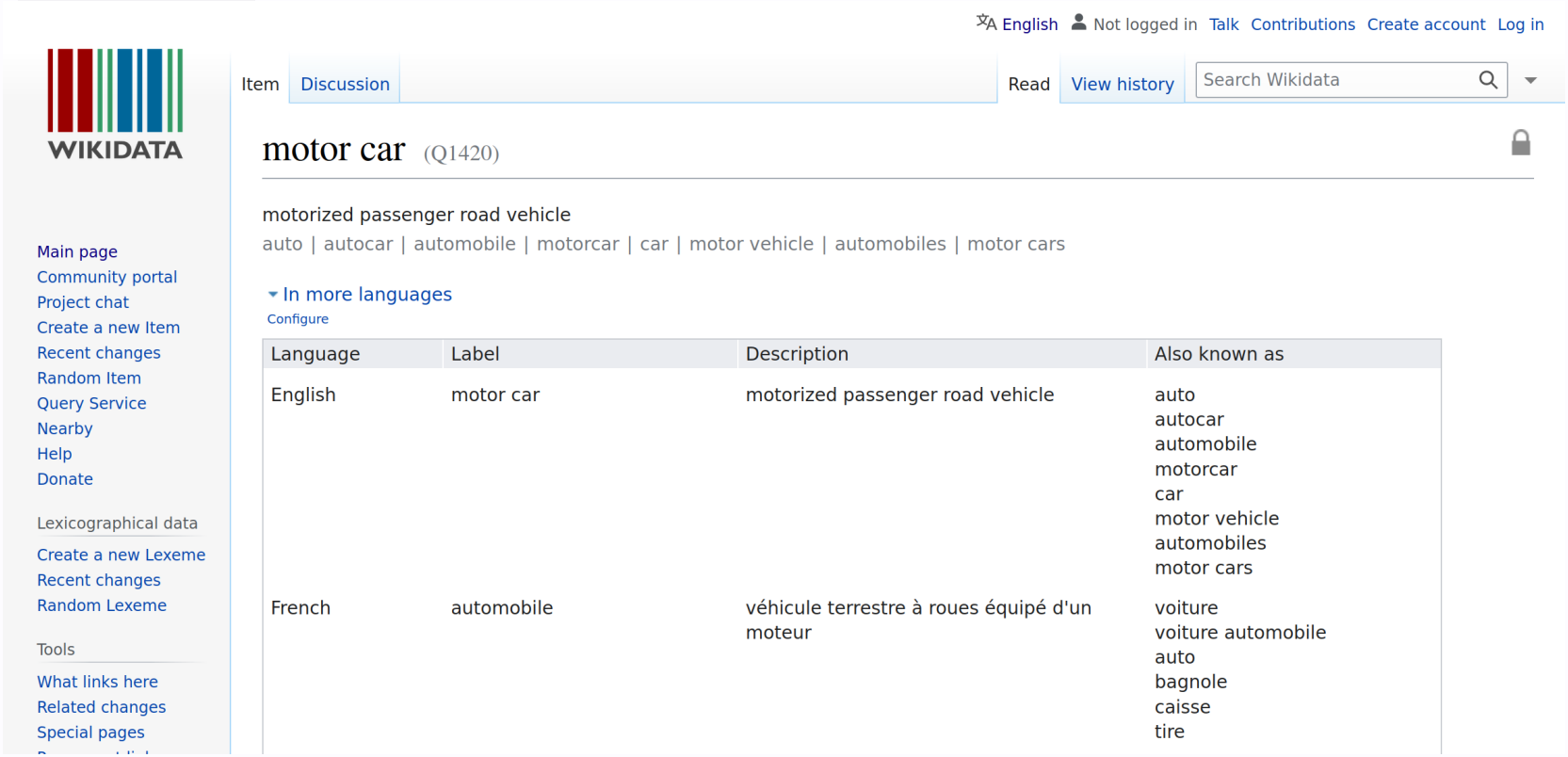
- But not a valid function so I searched for another function.

```
FILTER(STRSTARTS(?string1, ?string2))
```

- The second function I found to compare string was `STRSTARTS` .
- It will be useful to compare the name of a duck in german with the name of the cars.

Entities involved

- Entities were the `duck` entity and the `car` entity.
- I had to find the `car` entity and found the entity `Q1420` (see figure below).



English Not logged in Talk Contributions Create account Log in

WIKIDATA

Item Discussion Read View history Search Wikidata

Main page
Community portal
Project chat
Create a new Item
Recent changes
Random Item
Query Service
Nearby
Help
Donate

Lexicographical data
Create a new Lexeme
Recent changes
Random Lexeme

Tools
What links here
Related changes
Special pages

motor car (Q1420)

motorized passenger road vehicle

auto | autocar | automobile | motorcar | car | motor vehicle | automobiles | motor cars

In more languages
Configure

Language	Label	Description	Also known as
English	motor car	motorized passenger road vehicle	auto autocar automobile motorcar car motor vehicle automobiles motor cars
French	automobile	véhicule terrestre à roues équipé d'un moteur	voiture voiture automobile auto bagnole caisse tire

Entities involved (2)

- Only a few hundred names

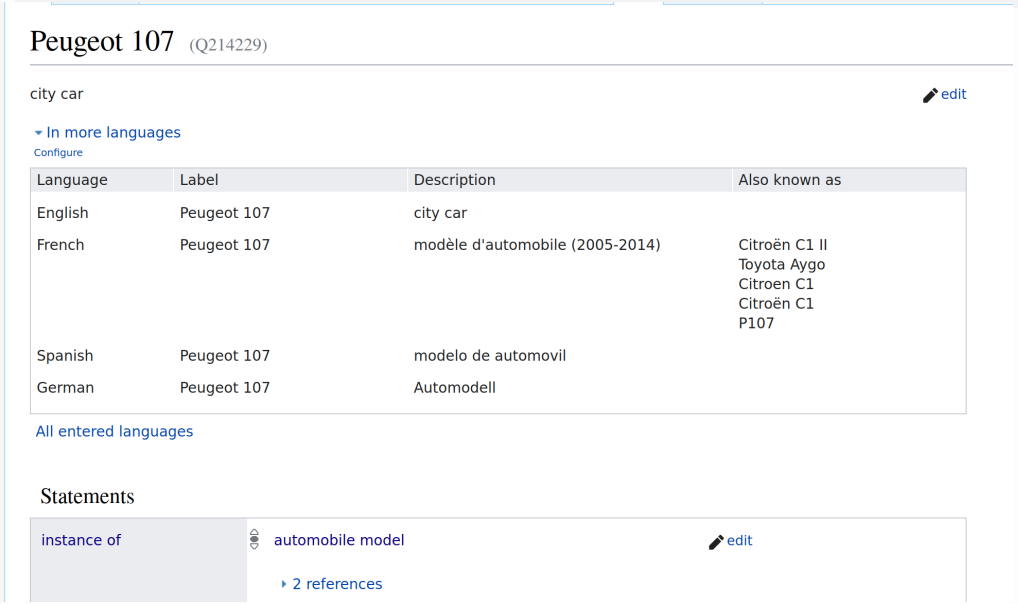
[16]:

```
%%rdf sparql --endpoint https://query.wikidata.org/sparql

# List all cars

SELECT ?carLabel
WHERE
{
            # instance of   # motor car
    ?car wdt:P31         wd:Q1420;
         rdfs:label ?carLabel.
    FILTER(LANG(?carLabel) = "en").

}
```

carLabel
Pontiac Bonneville@en
back-up car@en
Audi e-tron@en

Entities involved (3)

- I decided to use another entity.
- I tried typing my car model `Peugeot 107`.
- `automobile model (Q3231690)`.



Peugeot 107 (Q214229)

city car edit

In more languages

Configure

Language	Label	Description	Also known as
English	Peugeot 107	city car	
French	Peugeot 107	modèle d'automobile (2005-2014)	Citroën C1 II Toyota Aygo Citroen C1 Citroën C1 P107
Spanish	Peugeot 107	modelo de automovil	
German	Peugeot 107	Automodell	

All entered languages

Statements

instance of	automobile model	edit
	2 references	

Entities involved (4)

- I tried to use the `rdfs:label` of the entities to compare them
- Multiple unsuccessful results
- I found out that the car model `rdfs:label` did not include the car's **nickname**.
- I could use instead the `skos:altLabel` property of the entities that contained many more nicknames for the entity.

Filtering

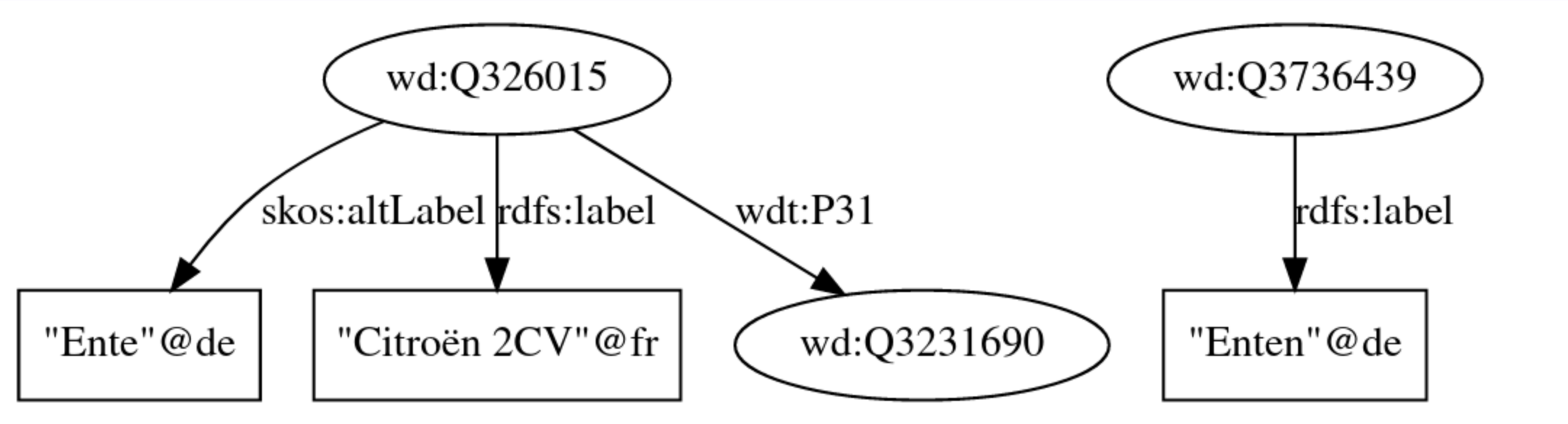
- I wanted to compare only the german `skos:altLabel` of the car with the german `rdfs:label`. So I searched and found the `LANG` function (see below).

```
-- Car
?model wdt:P31 wd:Q3231690;
    skos:altLabel ?modelAlias;
    FILTER(LANG(?modelAlias) = "de").

-- Duck
wd:Q3736439 rdfs:label ?duckLabel;
    FILTER(LANG(?duckLabel) = "de").
```

The query and result

```
CONSTRUCT {-- Hidden }
WHERE
{
    --      instance of     automobile model
    ?model wdt:P31          wd:Q3231690;
        skos:altLabel ?modelAlias;
        rdfs:label ?modelLabel;
    FILTER(LANG(?modelAlias) = "de").
    FILTER(LANG(?modelLabel) = "fr").
    FILTER(STRSTARTS(?duckLabel, ?modelAlias)).

    -- Duck
    wd:Q3736439 rdfs:label ?duckLabel;
        FILTER(LANG(?duckLabel) = "de").
}
```

The query and result (2)



- Here we can see that the car found is the `Citroën 2CV`.
- They both have a label and/or an alias wich begin with the string `Ente`.

Problem I want to solve

`With what other famous people did Barck Obama studied throughout his life ?`

Data source

- For this question I will use the same data source as the previous question
- WikiData.

Entities I need

- Barack Obama `Q76`
- Human `Q5`

Relations I need

- educated at `P69`

Listing the schools of Barack Obama

To list the schools of Barack Obama I used the following query:

```
%%rdf sparql --endpoint https://query.wikidata.org/sparql

-- With what other famous people did Barck Obama studied
-- throughout his life ?

SELECT ?schoolLabel
WHERE
{
    # Barrack Obama   # Educated At
    wd:Q76            wdt:P69          ?school.
    ?school rdfs:label ?schoolLabel;
        FILTER(LANG(?schoolLabel) = "en").
}
```

Listing the schools of Barack Obama (results)

schoolLabel
Harvard University
Columbia University
Harvard Law School
University of Chicago Law School
Occidental College
Punahou School
State Elementary School Menteng 01
Centaurus High School
King College Prep High School
Nelson High School
Noelani Elementary School

Listing 100 people who have studied at the same school as Barack Obama

```
SELECT ?personName
WHERE
{
    -- Barrack Obama   -- Educated At
    wd:Q76             wdt:P69          ?school.
    ?school rdfs:label ?schoolLabel;
    FILTER(LANG(?schoolLabel) = "en").
            -- instance of  -- Human
    ?person wdt:P31        wd:Q5;
        -- Educated At
        wdt:P69          ?school2;
        rdfs:label ?personName.
    FILTER(LANG(?personName) = "en").
    FILTER(?school = ?school2).
}
LIMIT 100
```

Problem encountered: Filtering with date

- The relation `educated at` had a qualifier `start time`
- I didn't know how to use it.
- So I had to look up WikiData documentation to find out how to use it.

```
    --! Educated At
    ?person p:P69 [
            --! Educated At
            ps:P69 ?school;
            --! Start Time
            pq:P580 ?date
        ];
```

Problem encountered: Filtering with date

```
FILTER(?school=?schoolObama
&& ((?startDate <= ?endDateObama
&&  ?endDate >= ?startDateObama)
|| (?startDateObama <= ?endDate
&&  ?endDateObama >= ?startDate)))
```

Problem encountered: Removing Barack Obama from the results

- **Barack Obama** and **Barack Obama** studied in the same school!!!
- I wanted to remove Barack Obama from the results
- So I had to find a way to remove him from the result. I tried to use the `FILTER`.

```
FILTER(?person != wd:Q76).
```

The query and result

```
SELECT ?nom ?schoolName ?startDate ?endDate
WHERE {
      wd:Q76 wdt:P69 ?school.
      wd:Q76 p:P69 [
        ps:P69 ?schoolObama;pq:P580 ?startDateObama;
        pq:P582 ?endDateObama].
      -- educated at
      ?person p:P69 [
        ps:P69 ?school;
        pq:P580 ?startDate;
        pq:P582 ?endDate
      ].
    FILTER(?school=?schoolObama && ((?startDate <= ?endDateObama &&  ?endDate
    ?person rdfs:label ?nom.
    FILTER(LANG(?nom) = "en").?school rdfs:label ?schoolName.
    FILTER(LANG(?schoolName) = "en").
    FILTER(?person != wd:Q76).
}
```

The query and result (2)

- **Michelle Obama** studied in the same school as Barack Obama in **Harvard Law School**.

nom	schoolObama	startDate	endDate
Ian Harrow	http://www.wikidata.org/entity/Q49088	1981-01-01T00:00:00Z	1984-01-01T00:00:00Z
Paul B. Reid	http://www.wikidata.org/entity/Q49088	1975-09-01T00:00:00Z	1981-08-01T00:00:00Z
Saeqa Vrtilek	http://www.wikidata.org/entity/Q49088	1982-02-01T00:00:00Z	1985-10-30T00:00:00Z
Michelle Obama	http://www.wikidata.org/entity/Q49122	1985-01-01T00:00:00Z	1988-01-01T00:00:00Z
Carol Raymond	http://www.wikidata.org/entity/Q49088	1982-09-01T00:00:00Z	1989-06-01T00:00:00Z
Joshua Bloch	http://www.wikidata.org/entity/Q49088	1978-01-01T00:00:00Z	1982-01-01T00:00:00Z

Thank you