Knowledge Graph

March 2021

1 Overview of knowledge graph

- Knowledge graph is a graphical representation of semantics in which each domains/entities are represented as nodes and connected by the edges with their relationship.
- DBpedia, Freebase, OpenCyc, Wikidata, and YAGO are the popularly used open source knowledge graph for cross-domain [1].
- Google coined the term Knowledge Graph. Besides, Google, IBM, Ebay, Facebook, Microsoft have their individual knowledge graph that are non-public [2].
- Knowledge graphs are created using different techniques [2] as follows:
  - **Manual creation** - Creating from base which is time and cost consuming as it involves crawling, segmentation, linking.
  - **Creation from Structured Sources** - Making use of available semantic web and adding knowledge related to the current domain. Example, DBpedia, YAGO, wikidata, CaLiGraph, BabelNet, ConceptNet, DBwik.
  - **Creation from Unstructured Source** - There is high possibility that the unstructured sources consists of rich information, such as text. Similar to web crawling. Example, Null, WebIsA.
- A general overview is given in Figure 6.

	DBpedia	YAGO	Wikidata	BabelNet
# Instances	5,044,223	6,349,359	52,252,549	7,735,436
# Assertions	854,294,312	479,392,870	732,420,508	178,982,397
Avg. linking degree	21.30	48.26	6.38	0.00
Median ingoing edges	0	0	0	0
Median outgoing edges	30	95	10	9
# Classes	760	819,292	2,356,259	6,044,564
# Relations	1355	77	6,236	22
Avg. depth of class tree	3.51	6.61	6.43	4.11
Avg. branching factor of class tree	4.53	8.48	36.48	71.0
Ontology complexity	SHOFD	SHOIF	SOD	SO

	Cyc	NELL	CaLiGraph	Voldemort
# Instances	122,441	5,120,688	7,315,918	55,861
# Assertions	2,229,266	60,594,443	517,099,124	693,428
Avg. linking degree	3.34	6.72	1.48	0
Median ingoing edges	0	0	0	0
Median outgoing edges	3	0	1	5
# Classes	116,821	1,187	755,963	621
# Relations	148	440	271	294
Avg. depth of class tree	5.58	3.13	4.74	3.17
Avg. branching factor of class tree	5.62	6.37	4.81	5.40
Ontology complexity	SHOIFD	SROIF	SHOD	SH

Figure 1: *General overview of knowledge graph [2].*

- Among DBpedia, Freebase, Wikidata, YAGO and OpenCyc, the major source of data for the first four is wikipedia, the way each graph link the entities are different. The later one is developed in Cyc language that is available both as commercial and opensource [1].
- Figure 6 depicts that wikidata consists of very high instances and classes. Comparing the overview given in [1] and [2] the wikidata has incorporated much information that indicates that the wikidata is getting updated continuously.
- Freebase acquired by Google in 2010 and transferred to wikidata in 2014 [3].
- Additionally the wikidata supports querying and multiple language porting compared to other opensource [1][4].
- Neo4J would benefit from better query using wikidata [4].
- **Wordnet** - One of the most well-known lexical databases for the English language, providing definitions and synonyms. Often used to enhance the performance of NLP and search applications [5].

2 Classes available in Knowledge graph regarding the project

The Aim of this session is to look for the amount of linkage and classes the vehicle, car, car crash, passive safety, EuroNcap possess in DBpedia, wikidata, Yago, BabelNet and LinkedGeoData.

2.1 Wikidata

- Following is the SparQl code [6] for the vehicle

```
prefix neo: <neo4j://voc#> #neo prefix to import it easily to neo4j
# Contruct the pair of 3 (Subject, Object, Predicate)
CONSTRUCT {
  ?item a neo:Category ; neo:subCatof ?parentItem .
  ?item neo:name ?label .
  ?parentItem a neo:Category; neo:name ?parentLabel .
  ?article a neo:WikipediaPage; neo:about ?item ;

}
WHERE {
  ?item (wdt:P31|wdt:P279)* wd:Q42889 .#Instance and subclass of the Q element
  ?item wdt:P31|wdt:P279 ?parentItem . #Link the classes
  ?item rdfs:label ?label .
  filter(lang(?label)="en")            #Extract labels in english for subclasses
```

```
?parentItem rdfs:label ?parentLabel .
filter(lang(?label)="en")            #Extract labels in english for main class


OPTIONAL {
  ?article schema:about ?item;
           schema:inLanguage "en";
           schema:isPartof <http://en.wkipedia.org/>.
  }
}
```

- When the following code is ran in the wikidata query Link., the query reaches timeout such that there could be lot of data to crawl.

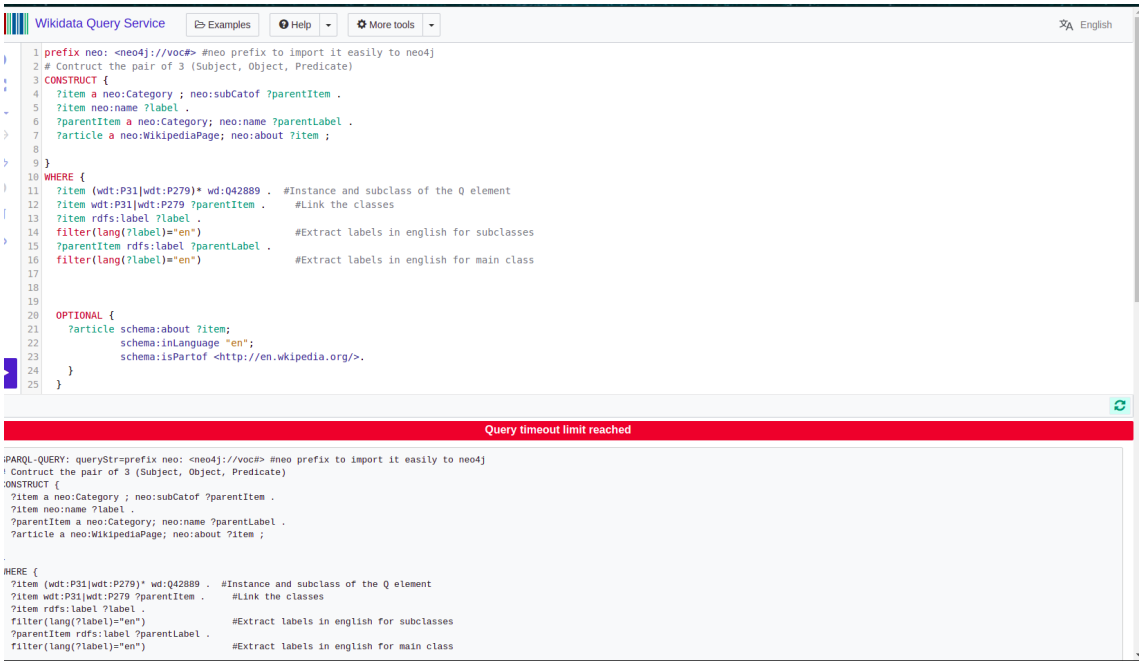

Figure 2: *Wikidata querying for vehicle.*

- When the query class is changed to Car (Q1420), produces better result but consists of car names like SUV, Audi etc no much useful information
- When the query class is changed to Car crash (Q61037771), passive safety (Q12043893) and EuroNcap (Q223400) produces very minimal results but not related to the project.

2.2 YAGO

- Yago query can be searched directly or can be done using SPARQL[7]

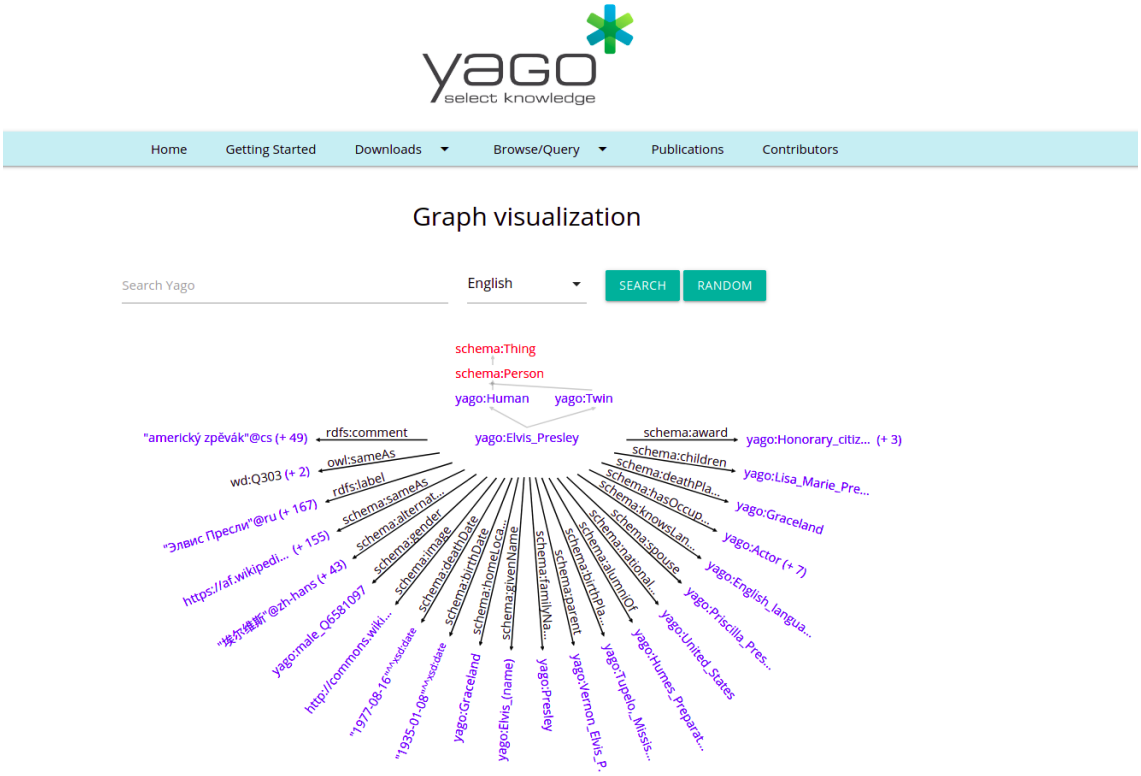

Figure 3: *YAGO knowledge graph.*

- For the search of vehicle the class linkage gives 5 things amoung them four are related to cars like Ferrari, Volkswagen, Austin, Fiat whereas the fifth one is breakdown but there is no subclasses in it.
- Search of car returns tons of car names nothing more than that.
- Car crash, passive safety and EuroNcap are not available itself.

2.3 BabelNet

- BabelNet is same as Yago; query can be searched directly [7]

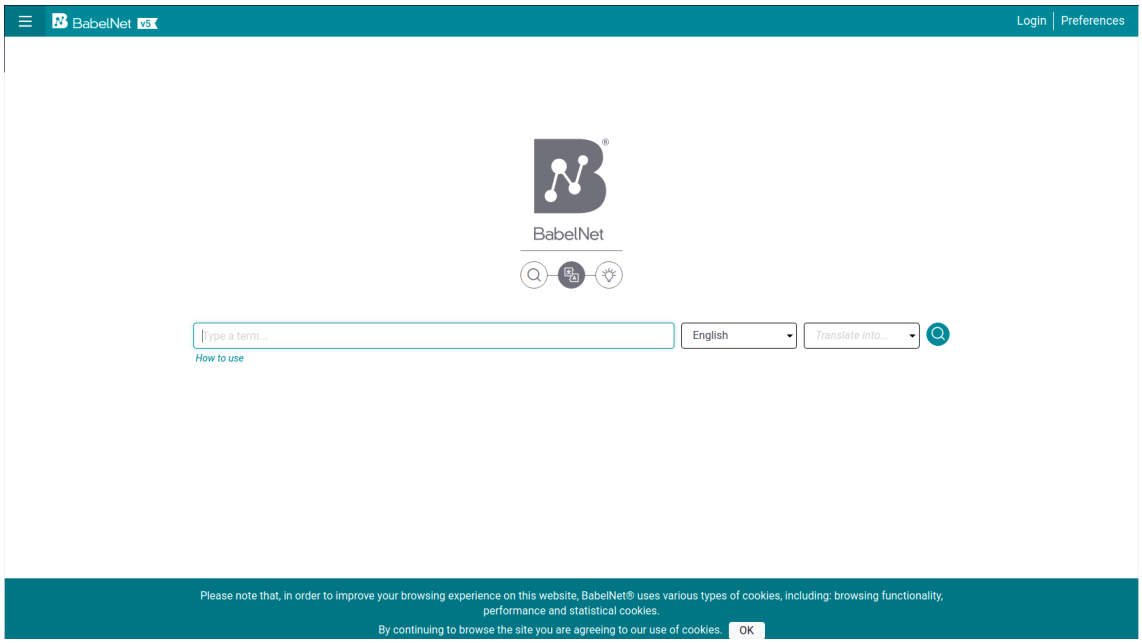

Figure 4: *BabelNet knowledge graph.*

- For the search of vehicle the class linkage gives a lot of information along with the car as one linkage
- Search of car returns tons of car names along with country, linkage to accident and car crash etc.
- Car crash, contains safety, EuroNcap within itself where EuroNcap consists of seatbelt, airbag, bumper which are certain information related to the project Link.

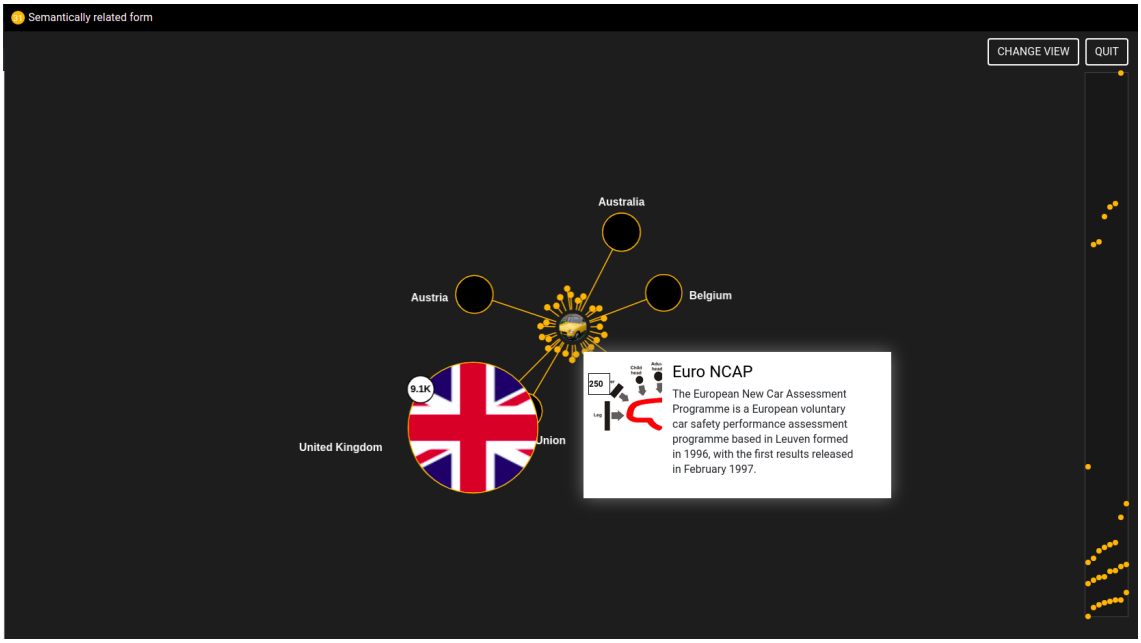

Figure 5: *BabelNet having EuroNCap.*

2.4 DBpedia

- DBpedia consists of SPARQL which is not much interactive like wikidata Link.

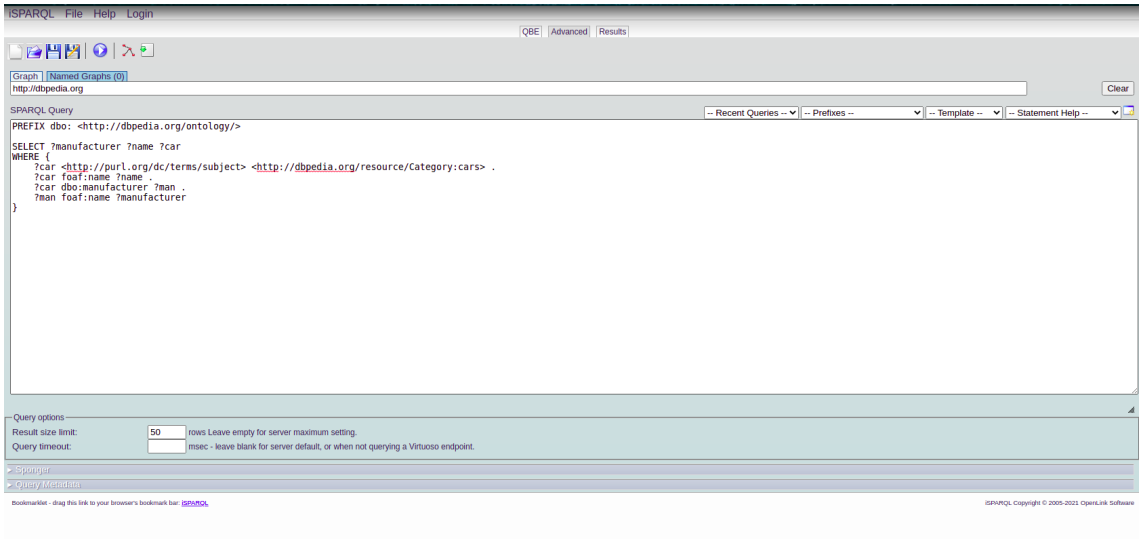

Figure 6: *DBpedia knowledge graph.*

- Consists of only Luxury_vehicles category doesn't contain anyother category we are looking for related to this section

3 code for scrapping

MATCH (b:'BEST IN CLASS CARS 2019') WHERE not exists((b)-[:YEAR]-¿())

CALL apoc.load.html(b.resultUrl, cars: ".car-model") YIELD value RETURN value

References

[1] M. Färber and A. Rettinger, "Which knowledge graph is best for me?," *ArXiv*, vol. abs/1809.11099, 2018.

[2] N. Heist, S. Hertling, D. Ringler, and H. Paulheim, "Knowledge graphs on the web - an overview," in *Knowledge Graphs for eXplainable Artificial Intelligence*, 2020.

[3] T. P. Tanon, D. Vrandečić, S. Schaffert, T. Steiner, and L. Pintscher, "From freebase to wikidata: The great migration," in *World Wide Web Conference*, 2016.

[4] D. Hernández, A. Hogan, C. Riveros, C. Rojas, and E. Zerega, "Querying wikidata: Comparing sparql, relational and graph databases," pp. 88–103, 10 2016.

[5] F. Nielsen, "Linking imagenet wordnet synsets with wikidata," *Companion Proceedings of the The Web Conference 2018*, 2018.

[6] Neo4j, "Tutorial: Build a Knowledge Graph using NLP and Ontologies," 2021. Accessed on: 2021-04-13. [Online].

[7] YAGO, "YAGO select knowledge," 2021. Accessed on: 2021-04-13. [Online].

[8] BabelNet, "BabelNet select knowledge," 2021. Accessed on: 2021-04-13. [Online].