Searching the web with Things

Pascal Molli and Hala Skaf-Molli
GDD Team
LS2N – Nantes University, France
{hala.skaf,pascal.molli}@univ-nantes.fr

Keywords: Semantic Web,Knowledge Graph, SPARQL Query Processing

Context and problem

Traditional web search engines search the web with strings. However, keywords search often returns many irrelevant documents, pushing users to refine their keywords list following a trial-and-error process. To overcome such limitations, major companies allowed searching for things, not strings[1] [44]. Asking for the age of "James Cameron" to your digital assistant locates in a **Knowledge Graph (KG)** a Person matching "James Cameron" where a property "age" is set to 67 years. Asking for the movies of James Cameron retrieves a list of Things representing Films linked to the Person "James Cameron". If searching for Things is amazing and delivers exact answers, it currently raises two major issues:

- **The search is performed over a Knowledge Graph and not the Web.** There may exist many answers or information that are only available on the web, and that are not part of the Knowledge Graph. A public KG such as DBpedia is built from Wikipedia, which is just a part of the web, not the whole web. Existing public KGs are fed with famous people, places, historical events, etc. Public KGs encode common knowledge by defining *KG entities*. What about ordinary people? Local events? Breaking news? Reviews or price of products? For instance, there is currently no way to retrieve Professors giving AI lectures at Master level in a French University. However, such information may exist in the web pages of Professors as *semantic annotations describing Web entities*. There is no fundamental difference between web entities and KG entities but they differ in the way they are produced and used. This led to two sets of entities that represent complementary knowledge and considering them as a whole allows building a richer web of data. According to webdatacommons, there are more than 18 billions web entities described by 80 billions facts defined over 14 millions websites. Public Knowledge Graphs define~~s~~ more than 28 £illions facts obtained over more than 650K datasets [45]. Although more and more data are being published either as KGs or microdata both still mostly "ignore" each other.
- **Only simple entity search can be performed, not complex queries.** Major search engines currently use KGs to answer Who, Where, When, What queries, e.g. "*Who directed Avatar*?". This usage is useful but very limited. As demonstrated by public KGs such as DBpedia or Wikidata, KGs can answer very complex queries such as "*Biomarkers that interact with proteins in human pathways*", or "*soccer players who were born in a country with more than 10 million inhabitants, who played as goalkeeper for a club that has a stadium with more than 30,000 seats, and whose club country is/was different from their birth country*". Such queries are not supported by major search engines, mainly for two reasons: (i) they have to be written in a query language that is too complex for an end-user, (ii) such queries may require a long time to be processed, so the search service cannot scale.

State of art

[1] https://blog.google/products/search/introducing-knowledge-graph-things-not/

Text	Keyword Search on Text	Structured Data Extraction from Text	Question Answering on Text
Knowledge Bases	Keyword Search on Knowledge Base	Structured Search on Knowledge Bases	Question Answering on Knowledge Bases
Combined Data	Keyword Search on Combined Data	*Semi-Structured Search on Combined Data*	Question Answering on Combined Data

Searching the web with things can be understood as a kind of semantic search [40], i.e. searching with "meaning". Semantic search can be classified following two dimensions: underlying data and search paradigm, as described in Figure 2.

- **Keyword search+text** corresponds to well-known web search engines, they are easy to use, but results can be noisy.
- **Structured search** corresponds to queries expressed with a query language. For example, the following SPARQL query **"select ?s where { ?s :has-profession :Scientist . ?s :birth-date "1967"}"** returns the scientists born in 1967. Structured search returns complete and correct results, but the query authoring can be complex.
- **Natural language search** often relies on queries starting with Who, What, Where, When, Why, How such as *"Who produced the Avatar movie?"*. This paradigm is completely natural for humans, but complex queries are difficult to phrase and queries can be hard to interpret due to the ambiguity of natural language.

In our context, we focus on semi-Structured Search on Combined data as proposed in Qlever [39]. However, QLever only considers text content and does not exploit semantic annotations. Considering semantic annotations extended with entity links coming from entity matching allows to improve the precision of the query and enrich the results by means of rich snippets.

Objectives of the PhD

The idea is to process queries in two steps:

1. First, extract collections of things from KGs as partial mappings of things where only one variable remains unbound, i.e. the web page where such things can be obtained.
2. Second, search on the web for which web pages contain such things. To retrieve web pages with things, web pages must be indexed with things and not keywords.

To power this idea, we aim to combine a preemptive KG query processing engine with an information retrieval engine based on an index of things. The KG preemptive engine is able to return things quickly, following orders specified in the query. Then web pages may be retrieved following simple rankings defined in the thing index.

The scientific challenge is to define a Multi-model SPARQL query engine able to support KG queries and information retrieval queries to answer structured queries.

This work is part of the ANR Project MeKaNo accepted in 2022.

Work program of MeKaNO

- **Indexing the web of things:** Web pages are currently indexed with words in a large inverted index. Our idea is to index web pages with things, i.e. URIs of public knowledge graphs. To perform this indexing, we can follow the approach of [39] or rely on information retrieval techniques. A special attention concerns the ranking of results retrieved by this index. Thanks to the presence of semantic annotation, we can define new ranking metrics taking into account the semantic attributes.

- **SPARQL+Web engine**: We plan to rely on SPARQL extensions to provide a unique interface for querying KGs and the web. But the processing combines fundamentally two engines: a preemptive SPARQL engine able to deliver KG partial mappings very quickly and a IR-entity based engine able to quickly search over the Thing-index with KG partial mappings. The KG preemptive engine is based on web preemption [GDD1,GDD2,GDD4,GDD5], a new scalable model for processing SPARQL queries. Web preemption slices the execution of SPARQL queries in time, ensuring that users see the progress of their queries every time quantum, e.g., every 100ms. A preemptable SPARQL server stops query execution after a quantum of time, and resumes the next waiting query. Stopped queries are returned to users with partial results and saved execution plans. Users may continue the query execution from where it was stopped by simply sending the saved execution plan to the preemptable server. Web preemption implements a pay-as-you-go model ensuring that only results observed by users are computed.
- **Benchmarking**: Performances are crucial in our context. In [39], QLever already compared the performances of the combined data approach and pure knowledge graph approach. The combined data approach delivered better execution time for queries. Our context is more complex than QLever as the text files contain semantic annotations. We have to ensure that the combined data approach still delivers better performances even in presence of semantic annotations.

Bibliography

[44] Natasha Noy, Yuqing Gao, Anshu Jain, Anant Narayanan, Alan Patterson, Jamie Taylor. Industry-Scale Knowledge Graphs: Lessons and Challenges. Communications of the ACM, August 2019, Vol. 62 No. 8.

[45] FERNÁNDEZ, Javier D., BEEK, Wouter, MARTÍNEZ-PRIETO, Miguel A., et al. LOD-a-lot. In : International semantic web conference. Springer, Cham, 2017. p. 75-83.

[40] Bast, Hannah, Buchhold Björn, and Elmar Haussmann. "Semantic search on text and knowledge bases." *Foundations and Trends in Information Retrieval* 10.2-3 (2016): 119-271.

[GDD1] T. Minier, **Hala Skaf-Molli, Pascal Moll**i (2019) SaGe: Web Preemption for Public SPARQL Query services. WWW Conference. The 2019 Web Conference, hal-02017155

[GDD2] Arnaud Grall, Thomas Minier, **Hala Skaf-Molli**, **Pascal Molli**. Processing SPARQL Aggregate Queries with Web Preemption. 17th Extended Semantic Web Conference (ESWC 2020), ⟨hal-02511819⟩

[GDD3] Alban Gaignard, Hala Skaf-Molli, Khalid Belhajjame (2020). Findable and reusable workflow data products: A genomic workflow case study. Semantic Web 11(5): 751-763 References

[GDD4] Julien Aimonier-Davat, **Hala Skaf-Molli, Pascal Molli.** Processing SPARQL Property Path Queries Online with Web Preemption. Extended Semantic Web Conference, Jun 2021, Hersonissos, Greece ⟨hal-03277623⟩

[GDD5] Julien Aimonier-Davat, **Hala Skaf-Molli, Pascal Molli**, Arnaud Grall, Thomas Minier (2022). Online approximative SPARQL query processing for COUNT-DISTINCT queries with Web Preemption. Semantic Web – Interoperability, Usability, Applicability, IOS Press, In press. ⟨hal-03563595⟩