Learning RDF pattern extractors from natural language and knowledge graphs -application to Wikipedia and the LOD (Poster)

Celian Ringwald, Fabien Gandon, Catherine Faron, Franck Michel, Hanna Abi Akl

Learning RDF pattern extractors from natural language and knowledge graphs - application to Wikipedia and the LOD

Célian Ringwald[1], Fabien Gandon[1,2], Catherine Faron[1], Franck Michel[1], Hanna Abi Akl[1,2]

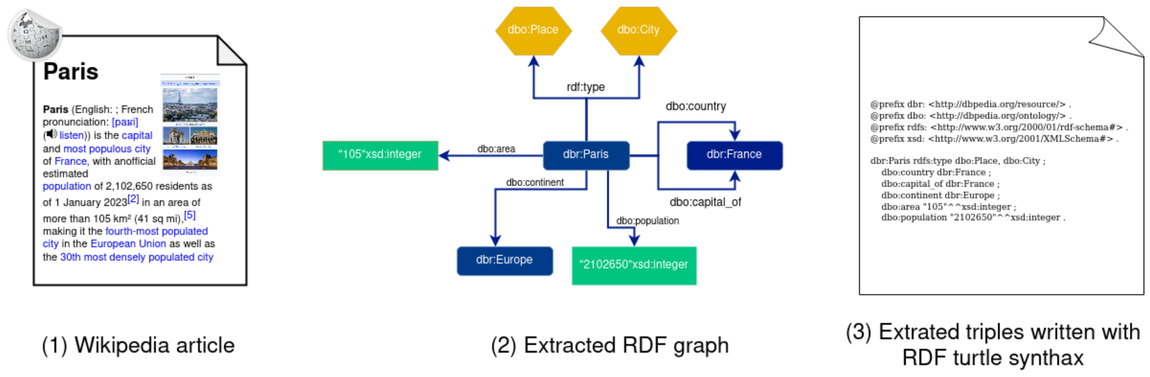
[1]Université Côte d'Azur, Inria, CNRS, I3S [2]Data ScienceTech Institute

Abstract

Whether automatically extracted from articles' structured elements or produced by bots or crowdsourcing, the open and linked data published by DBpedia and Wikidata now offer rich and complementary views of the textual descriptions found in Wikipedia. However, the text of Wikipedia articles still contains a lot of information that it would be interesting to extract for improving the coverage and quality of those bases. Until recently, relation extraction questions were solved by multiple-step processes. The latest improvement in deep learning and the development of large language models have shown their abilities in many downstream and complex tasks, and directly impacted the information extraction field. However, using and restricting these approaches to a given knowledge domain is still an open question. We are presenting here in more details our research questions, and how we are drawing a first overview of the current literature at the intersections of the knowledge graphs and language models fields.

Task description

The objective of our work is to extract and produce a set of RDF triples from a given input text following a predefined ontology and making references to entities known in a given knowledge base.



(1) Wikipedia article (2) Extracted RDF graph (3) Extrated triples written with RDF turtle synthax

Figure 1. Illustration of the abstract of the English Wikipedia article of Paris (1), its corresponding extracted RDF graph following the DBpedia ontology (2), and a textual description of the same RDF graph written with turtle syntax (3).

Previous works in relations extraction

- Initially handcrafted rule-based systems
  - → faced combinatorial complexity, semantic ambiguity
- Later elaborate pipelines emerged including multiple steps: named entity recognition, co-references and disambiguation, entity linking, relations extraction and classification.
  - → but long pipelines suffer from error propagation error.
- Today pre-trained language models have shown their ability to solve downstream tasks and generative models their ability to jointly extract entities and relations
  - → but they are not directly capable of solving a structured and reliable relations extraction.

An incremental methodology

1. From a single triple pattern, before generalizing to arbitrary graph patterns
2. From a subdomain of Wikipedia, before generalizing to any domain
3. From one natural language, before generalizing to more languages

Research questions

- **Datasets challenges** :
  - RQ1.1. How to support fact extraction relying on different document granularity?
  - RQ1.2. What is the best strategy to extract rare relations?
  - RQ1.3. How to represent a fact that spans several sentences and vice-versa?
- **Modelling challenges** :
  - RQ2.1. How to restrict language model to only facts that are supported by our text and our knowledge base?
  - RQ2.2. In our context, what is the best adaptation design for pre-trained models?
  - RQ2.3. Can we design an iterative procedure to produce consistent RDF triples ?

An enhanced tertiary review

As the landscape of the research fields drawn at the intersection of language models and knowledge graphs is very dynamic and quickly evolving, we chose to undertake a systematic review. This one will allow us to better understand the challenges relative to our task, the limits of current models, and follow the innovation made by the research community around these questions.

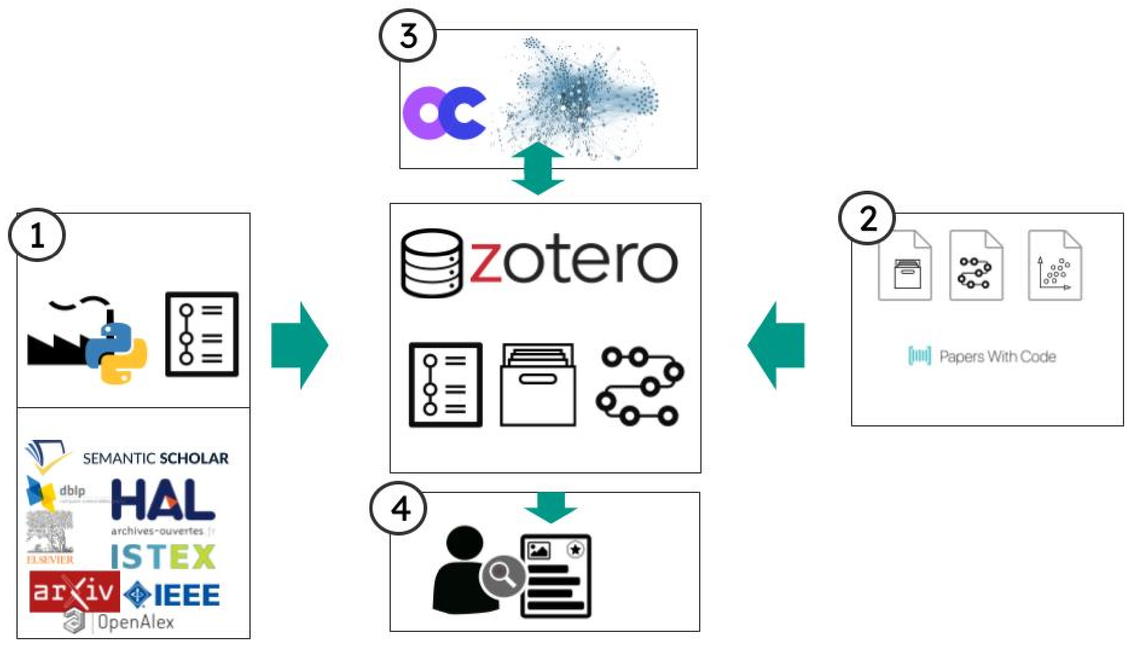


Figure 2. Our methodology in a nutshell

We kick-start our systematic review with a tertiary study (as illustrated in [Kitchenham and Charters 2007]: by looking for surveys and reviews already published and related to our task). We collected theses survey papers using digital libraries API. Then we extended the approach by :

- enriching and consolidating the tertiary review with the PaperWithCode dataset
- collecting the citation network of each papers via the use of OpenCitation API

A Zotero library was used throughout the process as a support in order to consolidate and curate our paper collection.

References

Kitchenham, B. and S. Charters (2007). *Guidelines for performing systematic literature reviews in software engineering*.