Learning Pattern-Based Extractors from Natural Language and Knowledge Graphs: Applying Large Language Models to Wikipedia and Linked Open Data

Célian Ringwald

Learning Pattern-Based Extractors from Natural Language and Knowledge Graphs: Applying Large Language Models to Wikipedia and Linked Open Data

Célian Ringwald[1]

[1] Université Côte d'Azur, Inria, CNRS, I3S
celian.ringwald@inria.fr

Abstract

Seq-to-seq transformer models have recently been successfully used for relation extraction, showing their flexibility, effectiveness and scalability on that task. In this context, knowledge graphs aligned with Wikipedia such as DBpedia and Wikidata give us the opportunity to leverage existing texts and corresponding RDF graphs in order to extract, from these texts, the knowledge that is missing in the corresponding graphs and meanwhile improve their coverage. The goal of my thesis is to learn efficient extractors targeting specific RDF patterns and to do so by leveraging the latest language models and the dual base formed by Wikipedia on the one hand, and DBpedia & Wikidata on the other hand.

Introduction

Whether automatically extracted from structured elements of articles or manually populated, the open and linked data published in DBpedia and Wikidata offer rich and structured complementary views of the textual descriptions found in Wikipedia. However, the unstructured text of Wikipedia articles contains a lot of information that is still missing in DBpedia and Wikidata. Extracting them would be interesting in order to improve the coverage and quality of these knowledge graphs (KG) since they have an important impact on all downstream tasks. This thesis proposes to exploit the dual bases formed from Wikipedia pages and Linked Open Data (LOD) bases covering the same subjects in natural language and in RDF, in order to produce RDF extractors targeting specific RDF patterns and tuned for a given language. Therefore, the main research question addressed in my thesis is:

RQ – *Can we learn efficient customized extractors targeting specific RDF patterns from the dual base formed by Wikipedia on one hand, and DBpedia and Wikidata on the other hand?*

Formally, let Db be a dual base, subset of $W \times G$ where W is the set of Wikipedia articles and G is the set of corresponding RDF graphs in DBpedia and Wikidata. The aim is to learn from this dual base an extractor $E_{Db} : W \rightarrow L; (t, S) \mapsto E_{Db}(t, S) = g$, where L is the LOD, t is an input text, S is a set of RDF patterns expressing constraints, and g is an RDF graph implied by t and valid against S.

This question is closely tied to the relations extraction (RE) task which consists of retrieving relations from unstructured texts. Until recently, the RE task was solved by complex pipelines including multiple steps (Martínez-Rodríguez, Hogan, and Lopez-Arevalo 2020). But this approach leads to error accumulations and propagation. However, the last advancements in NLP including pre-trained language models (PLM) have drastically improved the performance of many downstream tasks (QA, text summarising, translation...).

An incremental research plan

Following an incremental methodology, I intend to generalize the approach by relaxing one constraint at a time: starting from the generation of a single triple pattern before generalizing to arbitrary basic graph patterns. I will do so by addressing sequentially the following sub-questions:

SRQ.1 – *How to survey and follow the latest trends in PLM-based KG extraction?*
The landscape of the research field drawn at the intersection of language models and knowledge graphs is actually very dynamic and quickly evolving. Conducting a systematic literature review to closely follow it is crucial in this context.

Then, leveraging some of the latest techniques identified, the next research questions are set to find the currently best-performing approaches for a gradually more complex version of our task:

SRQ.2 – *Which aspects of the task formulation impact the generation of triples with datatype properties?*
The performance of the task learned on top of an existing language model crucially depends on the formulation of this task. In this sense, the choice of an RDF syntax or another for the extraction, and the content of the prompt given as input are sensitive parameters in order to take full advantage of the PLM. One of the first steps in this work will be to determine the combination leading to the best results by focusing only on the datatype properties.

SRQ.3 – *How to jointly extract datatype properties and object properties for a KG?*
The scientific community has recently underlined the "hallucination" problem of PLMs. In practice, this issue may affect a large proportion of the triples containing literals (e.g. attributes such as dates, measures, textual descriptions, etc.). These relations are defined in the OWL semantic web lan-

guage as datatype properties. In the literature, the research conducted until today was more focused on the extraction of object properties that link together two objects. I will have to propose a method to jointly extract both types of properties.

SRQ.4 – *How to support fact extraction relying on different document granularity?*
Relations can span different levels of a document (sentences, paragraphs, sections, etc.). The recent development of information retrieval techniques based on embedding seems to be a promising solution that must be adapted to our context.

SRQ.5 – *What is the best strategy to extract rare relations and under-represented instances of classes?*
In fact, the state-of-the-art models actually struggle with the under-representation of some facts. A lot of improvement must be made in this direction to be able to extract relations beyond those that are highly represented.

Preliminary results

To this day, I have built a tool for automating my literature review and conducted two preliminary experiments, in order to approach SRQ.1 and SRQ.2. This work and the results obtained from them will soon be formalized, extended, and submitted for publication.

A living and systematic review

The answer to SRQ.1 requires kick-starting a literature exploration from existing surveys related to a given task via querying several digital library APIs. The scientific corpora collected were extended by: (1) the dataset, the models related to our task, and the various metadata found on PaperWithCode; (2) the retrieval of the citation network of each paper using the OpenCitation API. The first run of the literature exploration allowed me to draw the current trends in RE. It suggests that the field is shifting from discriminative (encoder-based models derived from BERT) to generative modelling (based on encoder-decoder and decoder-only architectures). These models have the advantage of being flexible and enabling the conception of end-to-end systems. Moreover, they better handle overlapping triples extraction (Ye et al. 2022). I also underlined the fact that currently, very few initiatives already tried to use very large language models. Researchers were indeed more focused on the first generation of PLM. Finally, to my knowledge, no system is currently trying to perform semantic relation extraction where triples explicitly follow an RDF syntax.

A first focus on datatype properties

A first experiment has enabled me to tackle the first part of SRQ.2, by focusing on datatype properties, mainly affected by the hallucination problem. I have built the first dataset of relations based only on the English chapter of DBpedia and Wikipedia, restricted to entities of type `dbo:Person`, and focusing on the following relations: `dbo:birthDate`, `dbo:deathDate` and `rdfs:label`. I employed a SHACL shape to filter the graphs respecting the following criteria: an instance of `dbo:Person` must have a `dbo:birthDate`, a `rdfs:label` have a `dbo:deathDate`. On the example of REBEL (Huguet Cabot and Navigli 2021), I fine-tuned a BART model by giving as input the identifier of the entity followed by its Wikipedia abstract. The model was trained to generate RDF Turtle triples including the set of relations selected and found in DBpedia. A qualitative analysis of the errors allowed me to underline 3 sources of errors: (1) the fact is in the text but not in DBpedia, (2) the values in the text and in the DBpedia are different, (3) the fact is in DBpedia but not in the text.

I conducted a second experiment to understand if one syntax is easier to learn for a pre-trained model. In the literature, the choice of syntax is related to the "linearization process" where triples are serialized as a string: into the shape of a list, a tagged sequence. Until now, different methods have been investigated but they were not rigorously compared. For this reason, I extended the dataset of my first experiment to represent the triples according to seven different syntaxes: a simple list, a tagged sequence, a light Turtle syntax relieved of namespace prefixes, full Turtle, N-Triples, XML and JSON-LD. The results of the experiments showed us that BART was able to learn every syntax. But the fine-tuning of the model gave rise to heterogeneous learning times depending on the syntax before converging in terms of performance (in view of the recall, precision, and micro-F1 score metrics) : (1) the light-Turtle, the list, and the tag syntax were learned faster than the others. (2) These syntaxes are then followed by the Full-Turtle and JSON syntaxes. At the early stage of the training, the full-turtle recorded a slight inferiority in term precision, but at the same time, JSON-LD recorded a lower precision, giving the Full-turtle syntax an advantage in terms of the micro-F1 metric. (3) We have also pointed out that XML and N-Triples syntaxes are generally more difficult to learn as attested by the metrics.

Future works

Until the date of the Workshop, my research plan will be to continue to answer SRQ.2 and will extend to SRQ.3, by incrementally investigating new and more complex RDF graph patterns and integrating Wikidata to go beyond our current use of DBpedia. Aside from that, I will continue the analysis allowed by the systematic review launched for answering SRQ.1. I will then have another two years to cover SRQ.4 and SRQ.5. Depending on my advancement some extensions of the current plan could consider including approaches evaluating transfer learning and active learning.

References

Huguet Cabot, P.-L.; and Navigli, R. 2021. REBEL: Relation Extraction By End-to-end Language generation. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, 2370–2381. ACL.

Martínez-Rodríguez, J.-L.; Hogan, A.; and Lopez-Arevalo, I. 2020. Information extraction meets the Semantic Web: A survey. *Semantic Web*, 11: 255–335.

Ye, H.; Zhang, N.; Chen, H.; and Chen, H. 2022. Generative Knowledge Graph Construction: A Review. In *Proc. of the 2022 Conference on Empirical Methods in Natural Language Processing*, 1–17. ACL.