Well-written Knowledge Graphs Most Effective RDF Syntaxes for Triple Linearization in End-to-End Extraction of Relations from Text (Student Abstract)

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

Well-written Knowledge Graphs

Most Effective RDF Syntaxes for Triple Linearization in End-to-End Extraction of Relations from Text

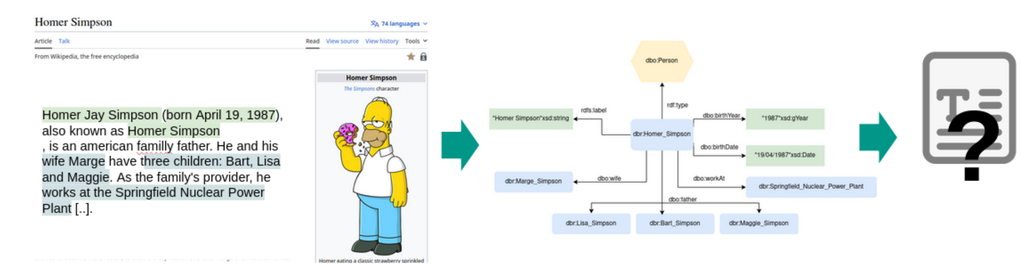
(Student Abstract)

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

Large generative language models recently gained attention for solving relation extraction tasks, notably because of their flexibility. This is in contrast to encoder-only models that require the definition of predefined output patterns. There has been little research into the impact of the syntax chosen to represent a graph as a sequence of tokens. Moreover, a few approaches have been proposed to extract ready-to-load knowledge graphs following the RDF standard. In this paper, we consider that a set of RDF triples can be linearized in many ways and we evaluate the combined impact of language model size as well as different RDF syntaxes on the task of relation extraction from Wikipedia abstracts.



Research question:

How does the choice of a specific RDF syntax impact the generation of triples using datatype properties?

Experimental setup

Dataset: `dbo:Person` instances from English DBpedia
+ SHACL Validation including the following datatype properties: `rdfs:label`, `dbo:birthDate`, `dbo:deathDate`, `dbo:birthYear`, `dbo:deathYear`.
+ Checking values in the Wikipedia Abstract.

Seven benchmarked syntaxes:

- Proposed by the W3C: *RDF-XML*, *JSON-LD*, *Ntriples*, *Turtle*.
- Proposed in the literature
  - *List* of relations :

```
((’Homer_Simpson’, ’type’, ’Person’),
(’Homer_Simpson’, ’label’, ’Homer Simpson’)...)
```

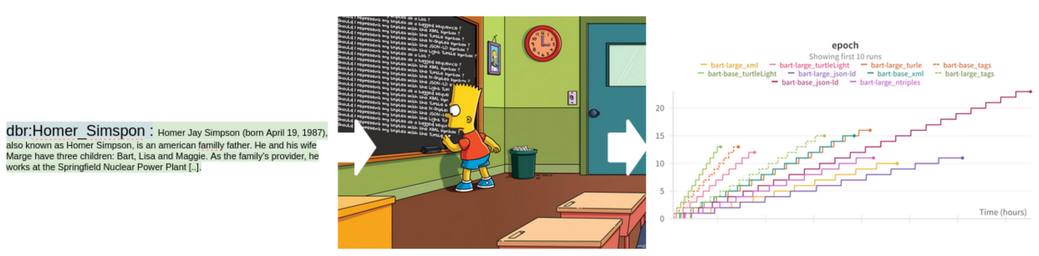
  - A tagged sequence:

```
<subj>Homer_Simpson<rel>type<obj>Person<et>
<subj>Homer_Simpson<rel>label<obj>Homer Simpson<et>...
```

- Proposed by our work: *Turtle Light*:

```
:Homer_Simpson a:Person ;
 :label "Homer Simpson";
 :birthDate "1987-04-19" ;
 :birthYear "1987.
```

Model: BART [Lewis et al. 2020] Base & Large FineTuned on each syntax. Train: 10 000 examples - Eval and Test: 3 000 examples. In Early stop mode: patience 5 training steps



Results & Discussions

Regarding the syntax:

- *Turtle Light* outperforms all the other syntaxes in every aspect
- *Tags* and *List* are easy to learn, but *List* needs more training epochs on BART-large;
- *Turtle* requires the same number of epochs for BART-base and BART-large;
- *JSON-LD* and *XML* are learnable syntaxes, that need more resources
- BART has difficulty learning *N-Triples*.

Overall: **The size of the model AND the chosen syntax impact the time needed to train an extractor of datatype relations but also impact the performance of the model itself.**

Figure 1. Micro-F1 performance of the models by size and syntax after a first epoch

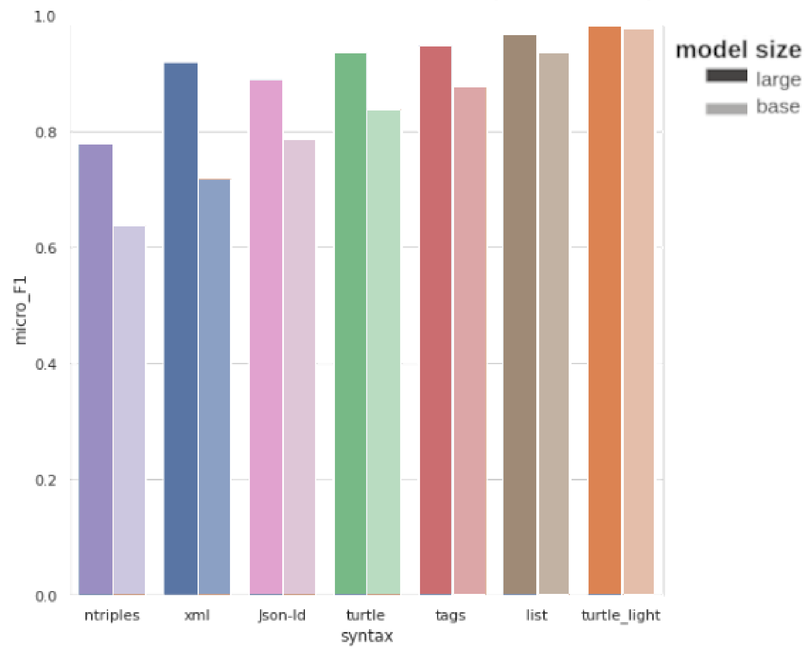
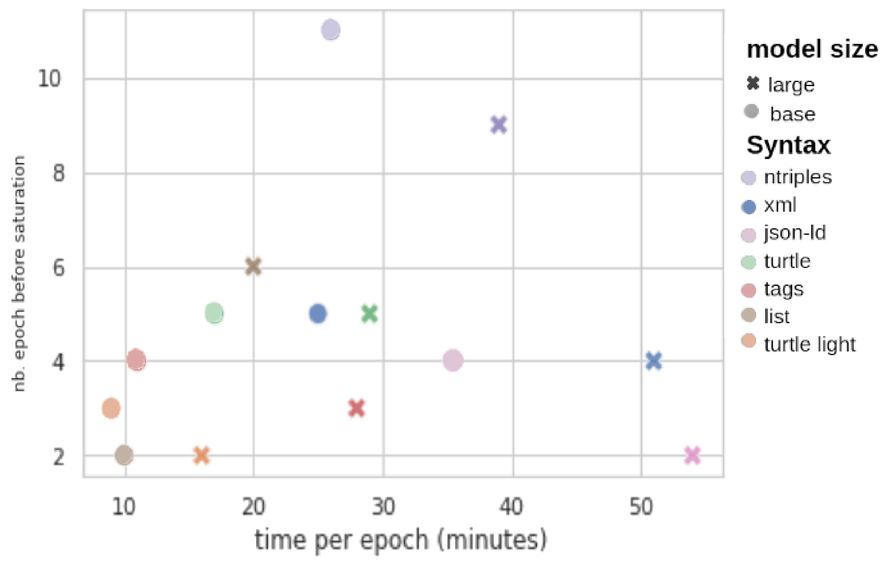


Figure 2. Number of epochs needed before reaching Micro-F1 saturation (> 0.9)



Future work

Improving the evaluation + Conducting a comprehensive analysis of the syntax variations + Adding the Syntax vocabulary into the tokenizer + Benchmarking other models from T5 and GPT families.

References

Lewis, M., Y. Liu, N. Goyal, M. Ghazvininejad, A. Mohamed, O. Levy, V. Stoyanov, and L. Zettlemoyer (2020). “BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension”. In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. ACL.