cqp4rdf: Towards a Suite for RDF-based Corpus Linguistics

Maxim Ionov[1][0000−0002−5631−1727], Florian Stein[1][0000−0001−6777−3356], Sagar[2][0000−0001−9781−9888], and Christian Chiarcos[1][0000−0002−4428−029X]

[1] Applied Computational Linguistics Lab, Goethe University Frankfurt, Germany
[2] Indian Institute of Information Technology, Sri City, India
{ionov,chiarcos}@informatik.uni-frankfurt.de,
flo@stein-software.com, sagar.r16@iiits.in

Abstract. In this paper, we present cqp4rdf, a set of tools for creating and querying corpora with linguistic annotations. cqp4rdf builds on CQP, an established corpus query language widely used in the areas of computational lexicography and empirical linguistics, and allows to apply it to corpora represented in RDF.
This is in line with the emerging trend of RDF-based corpus formats that provides several benefits over more traditional ways, such as support for virtually unlimited types of annotation, linking of corpus elements between multiple datasets, and simultaneously querying distributed language resources and corpora with different annotations.
On the other hand, application support tailored for such corpora is virtually nonexistent, leaving corpus linguist with SPARQL as the query language. Being extremely powerful, it has a relatively steep learning curve, especially for people without computer science background. At the same time, using query languages designed for classic corpus management software limits the vast possibilities of RDF-based corpora.
We present the middle ground aiming to bridge the gap: the interface that allows to query RDF corpora and explore the results in a linguist-friendly way.

Keywords: Linguistic Linked Data · Corpus linguistics · SPARQL · CQP.

1 Background

Corpora with annotations for features of morphology, grammar or semantics are fundamental to modern-day lexicography, linguistics, language technology and digital philology. Along with their broad range of uses, many different types of annotations emerged, leading to diverse and/or complicated data models, often with tool-specific data formats and a limited degree of interoperability. This interoperability challenge has long been recognized as an obstacle to scientific progress in the field, and has been the basis for developing language resource standards, e.g., the Linguistic Annotation Framework [12]. LAF implements the common insight that every form of linguistic annotation can be modelled as

a labelled, directed multi-graph. More recently, the application-specific stand-off XML formats previously developed in the language resource community to represent such graphs have been largely replaced by RDF-based data models, especially for annotations on the web or using web services [15, 11]. RDF is also applied to static language resources, e.g., machine-readable dictionaries [8] and terminologies [9], subsequently leading to the emergence of a linked data cloud of linguistic resources. Linked Data is a well-established paradigm for representing different types of data on the Semantic Web [1]. Linguistic Linked Open Data [3] describes the application of LOD principles and methodologies for modeling, sharing and linking language resources in various text- and knowledge-processing disciplines. For these areas, a number of benefits of LLOD and the underlying RDF technology over traditional representation formalisms have been identified [6]. Most notable for corpus linguistics, this includes:

1. Representation: linked graphs can represent any kind of linguistic annotation.
2. Interoperability: Different RDF graphs can be used together in a single query.
3. Ecosystem: broad support by off-the-shelf database technology.

The application of RDF and linked data technology to linguistic corpora is to be seen in this context and has been worked on for more than a decade, already [2, 10]. Despite the relevance, RDF is not well supported by existing corpus technology. Its current role in corpus linguistics is currently restricted to that of a publication format [14]. But aside from representing corpora, querying with Linked Data technology allows to query several resources in a single query, harmonizing different tagsets, creating intermediate annotations with annotation integration and flexible linking with other annotations, lexical resources (e.g. dictionaries or wordnets) and knowledge graphs as well as the linking (and querying) across concurrent annotations of the same text. Still, user- (i.e., linguist-) friendly interfaces to this technology are largely absent. This paper describes an effort to address this gap.

In our previous work, [4], we introduced a research methodology that uses these advantages in a typological linguistic study that relies heavily on corpora. Even though we found the approach valid, there was a downside: SPARQL is much more complex than traditional corpus query languages. This allows for more nuanced queries but makes the process of writing them more complicated: In addition to linguistic expertise required to know what to query, researchers need to have experience in SPARQL, or to work in tandem with semantic web professionals.[3]

In this paper, we introduce the new component of our methodological approach: `cqp4rdf`,[4] a collection of tools that allows querying corpora represented in RDF with CQP, a query language that is widely used for querying corpora with linguistic annotations, increasing the usability and visibility of Linked Data

[3] Total time spent on writing all the necessary queries for [4] was more than a week, it was done by a developer in tandem with a linguist.

[4] https://purl.org/liodi/cqp4rdf

corpora resources, making it possible to use them for corpus linguists unfamiliar with Semantic Web technologies. We briefly summarize the syntax of CQP with the additions we introduced to make it suitable for RDF data, and show the basic corpus manager interface that utilizes this.

2 CQP

CQP, the Corpus Query Processor, is a tool developed initially for the IMS Corpus Workbench [7]. It uses a query language which is also usually called *CQP*, or *CQP query language*.[5] Since its development, several major corpus management systems adopted it as a query language. The most prominent of them is *(no)SketchEngine* [13].

CQP is mainly intended for querying corpora with morphosyntactic annotations even though it has a possibility to query for segments if there are such annotations in corpora.

The query can consists of the three main types of expressions:

1. words and sequences of words filtered by attributes: `[]`, `[word="cat"]`, `[]+`, `[word="c.*"]`, `1:[]`
2. segment names: `<s/>`, `<p/>`
3. special constructions: `(not) containing` and `(not) within`
4. global conditions: `& 1.lemma = 2.lemma`

The first group matches consecutive words (tokens) that can be filtered using logical expressions that use token attributes, such as a part of speech or a lemma, e.g. `[word="comment" and pos="V"]`. They can be labeled, as demonstrated on the last example. The second group matches the whole structural segment. The presence of these segments depends on the annotations available in a corpus. The third group are operators between two sub-queries (which can contain one of these operators in turn). Finally, the last group allows to set constraints between tokens.

The non-standard feature we introduced to CQP are namespaces: all attributes and all segments should contain a prefix which correspond to a SPARQL prefix: `conll:WORD`, `nif:Sentence`. The list of recognised prefixes is defined in the configuration file. With this, there are no limitations on the vocabularies that are used in the corpus representation and there are no limitations on the list of properties that can be used to query corpora.

3 cqp4rdf

Currently, the tool set consists of a query conversion service, a backend which connects to a triple store to query for data and a frontend which provides a web interface. All these elements are packaged as a Docker container. There are

[5] Sometimes there is a confusion with *CQL*, which is another query language, still, some systems use *CQL* as the name for *CQP*

two steps required to set up the corpus: putting the data into a triple store and providing basic information about it in the YAML configuration file which stores configurations for all the corpora accessible for the `cqp4rdf` instance. The Docker container starts a triple store but it is possible to use an external one.

For the query conversion, we have implemented a eBNF grammar and used a parse tree of a query as a basis for further transformation to SPARQL. For this demonstration, it is adapted for corpora with CoNLL-RDF data model [5] which, in turn, relies on NIF [11], but adaptation to other data models can be achieved by modifying SPARQL templates which are used to generate queries.

On top of the conversion, we implemented an API that provides endpoints **/api/info** and **/api/query**. The former provides all the attributes of a token by its URI and the latter returns a list of results, transforming the input query and executing the SPARQL query returning a list of results for a specified page encoded in JSON. The output is limited to a number specified in the configuration file. Next sets of results can be retrieved with another request. The endpoint and the list of prefixes are specified in the configuration file as well.

This API is meant to be a backend, allowing to create tailored frontends for the specific needs or to adapt existing corpus management systems to use it. Currently, we implemented a minimalistic corpus interface[6] that allows to enter a CQP query and get a list of results in a KWIC[7] format (see Fig. 1). Every word in the output can be clicked to see the full information about this word.

Additionally, a user can construct the query using the user interface. The list of attributes and their possible values are specified in a configuration file. Note that there is no limitation on the attributes that can be used in a query, so users may still specify any additional attributes manually.

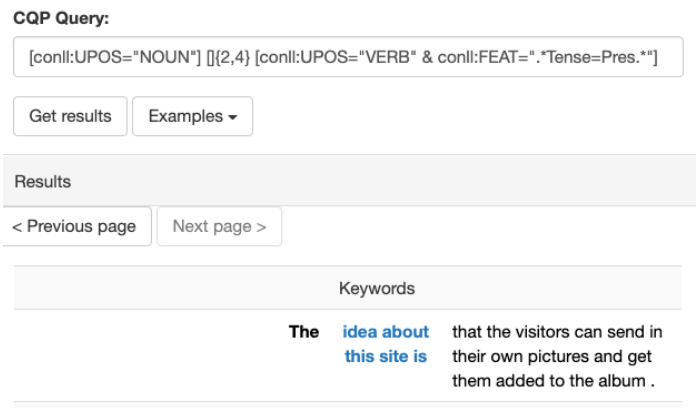

Fig. 1: Search results for a query that extracts contexts with a noun and a verb in present tense with 2 to 4 words in between.

[6] http://purl.org/liodi/cqp4rdf/ud
[7] Key-word in context

To compare the complexity of CQP and SPARQL queries, consider the query on the Fig. 1,[8] which returns contexts with a noun and a verb in a present tense with 2 to 4 words in between:

```
[conll:UPOS="NOUN"]
[]{2,4}
[conll:UPOS="VERB" &
 conll:FEAT=".*Tense=Pres.*"]
```

Listing 1.1: CQP version of the query

```
SELECT DISTINCT ?noun ?verb
WHERE
{
  ?w_1 a nif:Word ;
       conll:HEAD* ?sent ;
       conll:WORD ?noun ;
       conll:UPOS ?w_1_pos ;
       nif:nextWord{2,4} ?w_2 .

  ?w_2 a nif:Word ;
       conll:HEAD* ?sent ;
       conll:UPOS ?w_2_pos ;
       conll:FEAT ?w_2_feats ;
       conll:WORD ?verb .

  FILTER(REGEX(?w_2_feats,
    ".*Tense=Pres.*") &&
    ?w_1_pos = "NOUN" &&
    ?w_2_pos = "VERB")
}
```

Listing 1.2: SPARQL version of the query

The CQP representation, while may seem unfamiliar, is compact and intuitive, even for people with a limited knowledge of CQP. The corresponding SPARQL query is much more verbose and require knowledge of the underlying data model, whereas for the CQP query it is only required to know corpus attributes and a tagset (which is a prerequisite for using a corpus anyway).[9]

At the same time, corpus linguists can benefit from the underlying RDF format. For example, it is possible for a token to have multiple tags from different tagsets and combine them in the same query, or to have annotation provenance or other additional information stored and queried.

Even the minimalistic interface presented in this section allows to navigate through RDF corpora, increasing its usability, giving people unfamiliar with Semantic Web technologies the possibility to use the data quite efficiently.

4 Outlook

We showed our service for querying linguistic corpora represented in RDF with a common corpus query language, CQP. This approach increases the usability of RDF-based corpora, at the same time leaving the possibility to use the

[8] For brevity, we use non-normative SPARQL 1.1 Property Path (W3C Working Draft 26.01.2010), which is supported by some triple stores as an extension.

[9] This is, of course, not a problem of SPARQL but a result of using an intermediate conversion, which hides the data model under the hood.

whole arsenal of Semantic Web technologies when needed, for instance, to add intermediate annotations or to link elements to external vocabularies.

We see our work as a proof of concept implementation, and a basis for further studies. A number of open questions remain. First and foremost, the universality of such approach: How universal can be this service in terms of data models and vocabularies? Our next goal is to find the optimal middle ground between usability and expressiveness. Additionally, performance and scalability of the approach: corpus managers use highly optimized search mechanisms which require indexing data whereas in our approach the flexibility is prioritized over speed. Preliminary experiments with relatively small corpora shows that it does not cause problems but more benchmarks are required to test whether this can work for larger corpora.

References

1. Berners-Lee, T.: Linked data. Tech. rep., W3C Design Issue (2006)
2. Burchardt, A., Padó, S., Spohr, D., Frank, A., Heid, U.: Formalising multi-layer corpora in OWL/DL - lexicon modelling, querying and consistency control. In: Proc. IJCNLP-2008 (2008)
3. Chiarcos, C., Nordhoff, S., Hellmann, S. (eds.): Linked Data in Linguistics. Representing Language Data and Metadata. Springer (2012)
4. Chiarcos, C., Donandt, K., Sargsian, H., Ionov, M., Schreur, J.W.: Towards llod-based language contact studies. a case study in interoperability. In: Proc. LREC 2018 (2018)
5. Chiarcos, C., Fäth, C.: CoNLL-RDF: Linked corpora done in an NLP-friendly way. In: LDK. pp. 74–88 (2017)
6. Chiarcos, C., McCrae, J., Cimiano, P., Fellbaum, C.: Towards open data for linguistics: Lexical Linked Data. In: New trends of research in ontologies and lexical resources. Springer (2013)
7. Christ, O.: The IMS corpus workbench technical manual. Institut für maschinelle Sprachverarbeitung, Universität Stuttgart (1994)
8. Cimiano, P., McCrae, J., Buitelaar, P.: Lexicon Model for Ontologies. Tech. rep., W3C Community Report (2016)
9. Farrar, S., Langendoen, D.T.: A Linguistic Ontology for the Semantic Web. GLOT International **7**(3) (2003)
10. Frank, A., Ivanovic, C.: Building literary corpora for computational literary analysis-a prototype to bridge the gap between CL and DH. In: Proc. LREC 2018 (2018)
11. Hellmann, S., Lehmann, J., Auer, S., Brümmer, M.: Integrating NLP using linked data. In: Proc. 12th ISWC-Part II (2013)
12. ISO: ISO 24612:2012. Language Resource Management - Linguistic Annotation Framework. Tech. rep. (2012)
13. Kilgarriff, A., Baisa, V., Buta, J., Jakubek, M., Kov, V., Michelfeit, J., Rychl, P., Suchomel, V.: The Sketch Engine: ten years on. Lexicography (2014)
14. Mazziotta, N.: Building the syntactic reference corpus of medieval french using notabene rdf annotation tool. In: Proc. 4th Linguistic Annotation Workshop (LAW) (2010)
15. Sanderson, R., Ciccarese, P., Young, B.: Web annotation data model. Tech. rep., W3C Recommendation 23 February 2017 (2017)