Ordia
A Web Application for Wikidata Lexemes

Nielsen, Finn Árup

Published in:
The Semantic Web

Link to article, DOI:
10.1007/978-3-030-32327-1_28

Publication date:
2019

Document Version
Peer reviewed version

Citation (APA):
Ordia: A Web application for Wikidata lexemes

Finn Arup Nielsen
Cognitive Systems, DTU Compute, Technical University of Denmark

Abstract. Since 2018, Wikidata has had the ability to describe lexemes, and the associated SPARQL endpoint Wikidata Query Service can query this information and visualize the results. Ordia is a Web application that displays the multilingual lexeme data of Wikidata based on embedding of the responses from the Wikidata Query Service via templated SPARQL queries. Ordia has also a SPARQL-based approach for online matching of the words of a text with Wikidata lexemes and the ability to use a knowledge graph embedding as part of a SPARQL query. Ordia is available from https://tools.wmflabs.org/ordia/.

1 Introduction

The multilingual collaboratively editable and freely-licensed knowledge base Wikidata was set up in October 2012. On this website users can describe items and links between the items via properties, as well as add qualifiers and sources to support the individual claims. The Wikidata data—originally formatted in a nested JSON-like structure—is translated to a Semantic Web representation and continuously updated and made available via a SPARQL endpoint: The Wikidata Query Service (WDQS) and as such part of the Linked Open Data cloud.

In 2016, the Wikidata developers announced dictionary support in Wikidata, and in May 2018, Wikidata enabled the entering of basic data about lexemes and their forms. Later that year, Wikidata also switched on support for senses, and links to the Q-items from senses can be established. As the rest of Wikidata, the lexeme part of Wikidata is multilingual and ontological definitions in one language are available in other languages.

Below I will describe the Ordia Web application that takes advantage of the Wikidata lexeme data, aggregating the information via WDQS and presenting it on a website with added functionality in the form of lexeme extraction from a text and SPARQL integration of knowledge graph embedding information.

1 This work is funded by the Innovation Fund Denmark through the projects DAnish Center for Big Data Analytics driven Innovation (DABAI) and Teaching platform for developing and automatically tracking early stage literacy skills (ATEL).
4 The “ordinary” Wikidata items are referred to by an identifier consisting of the letter ‘Q’ and an integer, while the properties are identified by the letter ‘P’ and an integer. Lexemes are identified by the initial letter ‘L’.
2 Ordia Web application

Ordia is available from GitHub at http://github.com/fnielsen/ordia developed under the Apache 2.0 licenses. It may be cloned from that repository and run locally. The canonical homepage for the Web application is at https://tools.wmflabs.org/ordia/ under the Toolforge cloud service provided by the Wikimedia Foundation.

As a Web application and Python package, Ordia is heavily inspired from our other current Wikidata Web application projects: Scholia [3], cvrminer\(^5\) and Wembedder [2]: Ordia uses the Flask web framework together with Javascript and SPARQL templates in Jinja\(^6\) to dynamically build webpages. The constructed SPARQL queries are sent to the WDQS SPARQL endpoint and the responses are added to the generated HTML pages, either with HTML embedding or via Javascript and the DataTables library.\(^7\) The library provides means for sorting table rows and for drill down via a search field. The SPARQL queries used to generate the tables in Ordia are all linked from an anchor in the lower left corner of the tables, making a SPARQL-knowledgeable user able to inspect and modify the queries.

Ordia creates separate pages for Q-items, lexemes, forms and senses, and makes panels with tables on each of them. Fig. 1 shows an example for a lexeme. Ordia uses a URL scheme for Q-items inspired from Scholia’s notion of aspects and shows aspects for language, lexical category, grammatical features, properties and references, e.g., the link /language/Q809 will show Polish (Q809) lexemes, while /Q809 shows Polish as a semantic concept in its own right. Some of the aspects show graphs for the ontology, e.g., the page for noun as a lexical category at /lexical-category/Q1084

For searching after lexemes and forms, Ordia uses the MediaWiki API of Wikidata: The user types in a search in the Ordia interface and Ordia makes an API call to Wikidata and presents the results in the Ordia interface.

\(^5\) Descriptions of cvrminer at https://tools.wmflabs.org/cvrminer/ has not been published. The Web application displays information about organization as listed in Wikidata.

\(^6\) http://flask.pocoo.org/ and http://jinja.pocoo.org/

\(^7\) https://datatables.net/.
**Wembedder**, a Wikidata-based knowledge graph embedding Web service [2], works with a simplified RDF2Vec approach implemented with Gensim’s word2vec model [5,8,1]. As Ordia, Wembedder runs as part of Toolforge. The current implementation only handles the Q-items and properties of Wikidata, not lexemes, forms nor senses. The only functionality implemented in the Wembedder Web service so far is a most similar service that returns the most similar items and properties based on a query item. Wembedder has no SPARQL endpoint capability, so federated SPARQL queries cannot be made. Instead Ordia calls the REST interface of Wembedder via a Javascript Ajax call from the server side and formats the received JSON with Wikidata identifier and similarity values as two-tuple values for the SPARQL VALUES construct. The VALUES construct is then interpolated into a SPARQL template and sent off to WDQS with the response formatted in Ordia in a table with the DataTable library.

Ordia uses the Wembedder queries on pages for Q-items, where a table displays related Q-items sorted according to similarity and augmented with information from the lexeme part of Wikidata. Fig. 2 shows an example of the output on the page for the concept *Thursday* corresponding to the page [https://tools.wmflabs.org/ordia/Q129](https://tools.wmflabs.org/ordia/Q129), where the top panel displays lexemes for languages linked (e.g., jeudi, Donnerstag, Thursday) and the lower panel shows the result from WDQS with Wembedder-included results. Here Wednesday and Saturday are the most related concepts to Thursday.

The text-to-lexeme facility in Ordia at [https://tools.wmflabs.org/ordia/text-to-lexemes](https://tools.wmflabs.org/ordia/text-to-lexemes) enables the user to write a short text into an HTML text area on the client side, and send it off to Ordia. Ordia then makes a simple sentence detection and lowercases the first letter of the sentences before word tokenization with a simple regular expression pattern. Identified words are interpolated into a WDQS query via the VALUES keyword to search for matching forms, and the response from the SPARQL endpoint is shown in a table in the Ordia interface. The

---

8 [https://tools.wmflabs.org/wembedder/](https://tools.wmflabs.org/wembedder/)
Fig. 3 displays the result after the Danish sentence “Regeringen spiser grønne æbler om vinteren” (“The government is eating green apples during winter”) has been submitted to Ordia. The result of the WDQS query here shows the word and—if matched—the form, lexeme, lexical category, lexical feature, sense and image associated with the sense. If a word matches several forms in a language, they are all shown, i.e., no word sense disambiguation is performed. Ordia’s text-to-lexemes responds within seconds. Usually Ordia responds with the HTML within 300 milliseconds for a sentence like the above. The SPARQL query sent by the client to WDQS completes typically between 1.5 and 2 seconds after the user submitted the original query. The further download and rendering of the images from Wikimedia Commons—as shown in Fig. 3—may take an extra second. If the SPARQL query does not find a matching form, a link is created to Ordia’s search page, which links further on to lexeme creation to ease the setup of new lexemes.

3 Discussion

I have shown Ordia, a Web service that uses the WDQS SPARQL endpoint to build a site with lexicographic data from Wikidata. Compared to the Wiktionary-based DBnary [6], Ordia needs no extractor and presents the lexicographic information graphically and up-to-date via WDQS as changes occur in Wikidata.

The conceptual choices that has been made in designing Ordia are: 1) A user should easily be able to perform powerful SPARQL queries by navigating the Ordia interface and without using any knowledge of SPARQL; 2) the URL pattern for each page should be easy to understand and predict; 3) Each page should link to other pages and in such a way let the user discover new lexemes, concepts, forms etc., and 4) the interface should use graphics when possible, e.g.,
graphs of word and concept relations and for displaying images associated with senses of words.

Other Wikidata lexeme Web applications beyond Ordia are available: Lucas Werkmeister has created Wikidata Lexeme Forms which enables easy HTML-form-based set up of lexemes and their lexical forms for a range of languages. In November 2018, Werkmeister found that this tool has been used for the creation of 10'827 lexemes out of a total of 37'886.