

# New developments in Lexonomy

Adam Rambousek<sup>1,2,4</sup>, Miloš Jakubiček<sup>1,2</sup>, Iztok Kosem<sup>3,4</sup>

<sup>1</sup>Faculty of Informatics, Masaryk University, Brno, Czech Republic

<sup>2</sup>Lexical Computing, Brno, Czech Republic

<sup>3</sup>Centre for Language Resources and Technologies, University of Ljubljana, Slovenia

<sup>4</sup>Jožef Stefan Institute, Ljubljana, Slovenia

E-mail: rambousek@fi.muni.cz, milos.jakubicek@sketchengine.eu, iztok.kosem@cjvt.si

## Abstract

This article describes new developments and enhanced features in the open-source web application for dictionary writing, Lexonomy. Since its introduction in 2017, a growing number of users and organisations have chosen Lexonomy to edit their dictionaries. We describe the motivation and process of the source code refactoring to Python programming language. Next, we provide details on integration with the Sketch Engine corpus manager. We also cover the completely new feature of dictionary linking, both as a graphical interface for users, and API to include Lexonomy in the process of automatic dictionary linking. Finally, the article describes the new functionality needed for Lexonomy integration within the ELEXIS project processes. Furthermore, we provide usage statistics on users and dictionaries they create.

**Keywords:** Dictionary editing; Dictionary writing system; Lexicographic tools; XML; Corpora connection

## 1. Introduction

Lexonomy (Měchura et al., 2017) is a free, open-source, web-based dictionary writing system. Since its introduction in 2017, it is used by a growing number of users and organisations. The publicly available installation at [www.lexonomy.eu](http://www.lexonomy.eu) is currently used by over 2,700 users who created over 5,000 dictionaries.

Lexonomy was selected to be part of the ELEXIS (Krek et al., 2018) project infrastructure, providing the primary tool for dictionary creation, storage, and browsing. Thanks to this, the number of users and their dictionaries increased significantly, which led to two groups of updates to Lexonomy. Integration into ELEXIS brought new feature requests from various project partners. Furthermore, we had to address performance issues for a larger amount of data and users.

The following chapters present new updates and features in Lexonomy since 2018.

## 2. Improved scalability

Originally, Lexonomy was developed in Node.js<sup>1</sup> at the backend side and HTML+JavaScript on the client-side. While the Node.js server provided a connection to the database, core functionalities, and application interface, HTML webpages enriched with JavaScript provided a graphical user interface. To store metadata about users and dictionaries, and dictionary entries, Lexonomy uses the SQLite database<sup>2</sup>. Each dictionary is stored in a separate database file. One of the benefits is working directly with the database file, e.g., using dictionary templates for various projects or backup.

As the number of users and dictionaries in the system grew, we experienced performance issues and long response times when users searched in their dictionaries. After profiling all parts of the application, we identified the handling of concurrent database access requests

<sup>1</sup> <https://nodejs.org/en/about/>

<sup>2</sup> <https://www.sqlite.org/>

to be the main cause of the issue. When many users at once searched for entries or imported dictionary data, Node.js server kept database queries in a queue and processed them one by one. This means that one complex database search or import of extensive data into a dictionary may slow down the response time for other users.

At the same time, more developers wanted to participate in Lexonomy, and the issues with Node.js meant that they had to wait before they were able to join the team.

We thus decided to refactor the code of the backend part of Lexonomy. After considering the pros and cons of several programming languages, we selected Python as the best option. From the beginning, we addressed performance by using a multi-threaded environment and running time-consuming tasks (e.g., dictionary import) as background jobs.

After we deployed the refactored backend on the Lexonomy server, Lexonomy could smoothly handle dictionaries of millions of entries. Users only noticed the better performance of Lexonomy, as the graphical user interface was not changed and it still uses the same HTML templates with JavaScript. For developers, the Lexonomy source code is now smaller and more transparent, and they do not need to repeat the same code several times (e.g., checking user access rights).

### **3. Closer integration with Sketch Engine**

Lexonomy may still work as a standalone tool that can be installed locally on anybody's desktop. It can also be easily coupled with the (No)Sketch Engine corpus management system (Kilgarriff et al., 2014) to get access to corpus content. Connection with the Sketch Engine was extended to provide more options and a better user interface.

The first option is to retrieve the corpus data directly while working in the dictionary editor. For each dictionary, users can select which corpus to use and which elements in the entry structure correspond with different corpus data (examples, collocations, thesaurus items, or definitions). When editing an entry, users will see the Sketch Engine icon on the right elements. After clicking the icon, they may run a CQL query and select which results to include, see Figure 1 for example. The data will be copied to the dictionary entry structure where users can post-edit them. As a default, sketchengine.eu is accessed. However, users may specify their own installation of (No)Sketch Engine.

And from the other side, it is possible to create a new dictionary and fill it with data from the Sketch Engine interface. Users will start in the Sketch Engine and its OneClick Dictionary tool (Kilgarriff & Jackson, 2013). Depending on language support and user selection, the process generates a headword list with part-of-speech labels, provides candidates for example sentences, collocations, synonyms, or definitions. Subsequently, all the data are pushed into Lexonomy, where the new dictionary is created. Users are able to extend or edit the dictionary during the post-editing phase, thus saving time.

### **4. Single sign-on**

To make registration and authentication more comfortable for users, Lexonomy provides the option to log in via the Sketch Engine application. Thanks to this integration, users

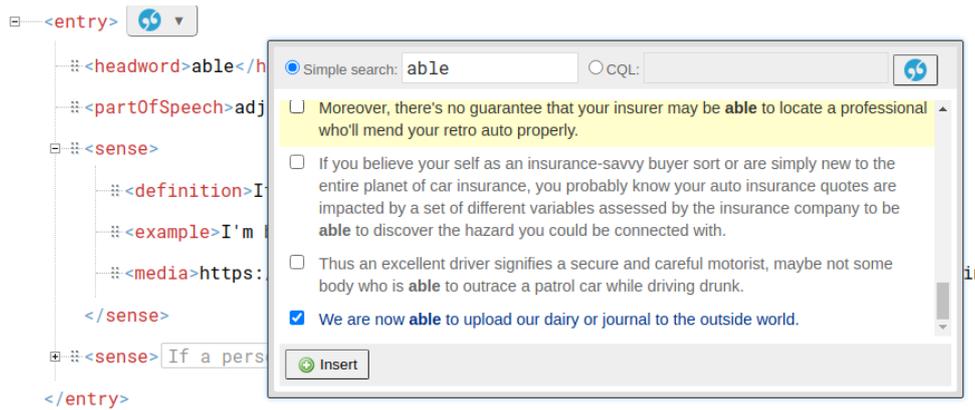


Figure 1: Connection with the Sketch Engine, searching for example sentences.

are able to log in to Lexonomy with easy single sign-on through the worldwide eduGAIN research network<sup>3</sup> and other institutions.

## 5. Integration with Elexifier

Lexonomy was selected as a primary tool for dictionary creation and editing in the ELEXIS project. Apart from dictionary editing, Lexonomy is the base for Elexifier (McCrae et al., 2019), a tool that is designed to digitise printed dictionaries in PDF or XML format. Utilising the option to change the default Lexonomy entry editor with custom JavaScript and XSLT code, Elexifier developers created their own entry editor for annotation of dictionary data in PDF files.

## 6. Dictionary linking

Lexonomy was selected as the dictionary storage in the ELEXIS project, where available dictionaries will be interlinked. To support this task and other scenarios where users need to connect dictionaries, Lexonomy was extended with the general mechanism for dictionary linking.

### 6.1 Manual linking

The linking mechanism in Lexonomy supports links between any entry elements in any dictionary. As a first step, users have to specify which entry elements should serve as the link point and how each element is identified. For example, *entry* may serve as a link point and each entry is uniquely identified with *(headword + PoS)*, or *definition* may be used as a link point and each definition is uniquely identified with *(headword + PoS + sense number)*.

When users are editing an entry, they have the option to add or view links at corresponding entry elements. When they want to add a new link, they select the target dictionary, choose which element to use in the target entry, and search for a particular link target. Source and target elements may be on a different level in an entry structure. For example, it is

<sup>3</sup> <https://technical.edugain.org/status>

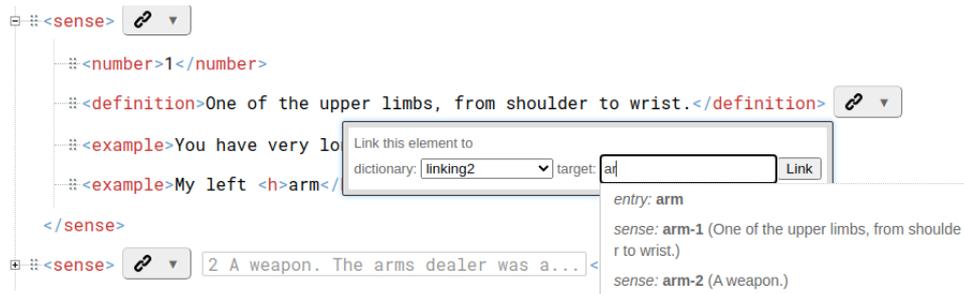


Figure 2: Creating link between *definition* and searching for target element (*entry* or *sense*).

**Herrgott**

*Gott, und zwar auch i. S. v. Christus und Hostie, als ein Begr. gedacht, nicht da Volke nicht gebraucht, sondern entw. «de' lieb Gott» oder dann eben «de' Herr*

**Incoming links**

44126\_1 ← [lexis-oeaw-schranka : Herrgott : sense : 25621\\_1 \(1\)](#)

Figure 3: Example of link information in an entry preview.

possible to create a link between full entry and one definition. See Figure 2 for an example of link creation and searching for the target of the link.

When browsing the dictionary, links are also displayed in the entry preview (see Figure 3). To provide a general overview, Lexonomy also displays the complete list of links for the dictionary (see Figure 4).

**6.2 Automatic linking**

For integration with automatic linking tools, Lexonomy provides API interface to work with the cross-links. As of now, the NAISC tool (McCrae & Buitelaar, 2018) is available for automatic linking directly from Lexonomy. Although the process was developed with the NAISC tool, it may be easily extended to work with other tools.

The process of automated link creation uses the following steps:

- user selects source and target dictionary,
- both dictionaries are converted to the OntoLex RDF format required by NAISC,
- NAISC detects the links,
- output from NAISC is converted to the internal Lexonomy format and stored in the database,
- links are available, and users may post-edit the results in Lexonomy editor.

As an input, NAISC requires files in the OntoLex RDF containing headword, part-of-speech, and definitions texts for each entry. Since we anticipate many dictionaries with various entry structures, users may not be able to configure linking elements

The screenshot shows a web interface for 'elexis ZRC SAZU JSV'. At the top, there are buttons for 'Edit', 'Configure', and 'Download'. Below this, the section 'Outgoing links' is displayed. A sub-section titled '→ elexis-zrcsazu-pletersnik' contains a list of 25 entries. Each entry consists of a blue hyperlink followed by a sense number in parentheses, an arrow pointing to another blue hyperlink, and another sense number in parentheses. For example, the first entry is 'lorber (3825\_1) → lorber (sense 30798\_1)'. The list includes various words like 'truplo', 'nadražiti', 'počasi', 'enak', 'coprati', 'korar', 'skriven', 'purman', 'odlašati', 'raznesen', 'zvezati', 'nerodovit', 'oblegati', 'premilostljiv', 'oves', 'izdati', 'zakleti', 'leški', 'natočiti', 'pastirica', 'tam', 'drokniti', and 'ris 1'.

Figure 4: Example of dictionary overview of all available links (from the *JSV* dictionary to the *Pleteršnik's* dictionary, linking between senses of both dictionaries).

beforehand for each dictionary. In such a case, Lexonomy tries to guess the entry structure to provide all the data for NAISC – starting with the TEI-Lex0 entry structure, followed by several common elements for headwords and definitions (see Figure 5 for an example of OntoLex RDF export).

```
<http://lexonomy.elex.is/elexis-oeaw-schranka#AbkrageIn_25132> <http://www.w3.org/1999/02/22-rdf-syntax-ns#type> <http://www.w3.org/ns/lemon/ontolex#LexicalEntry> .
<http://lexonomy.elex.is/elexis-oeaw-schranka#AbkrageIn_25132> <http://www.w3.org/2000/01/rdf-schema#label> "AbkrageIn"@de .
<http://lexonomy.elex.is/elexis-oeaw-schranka#AbkrageIn_25132> <http://www.w3.org/ns/lemon/ontolex#sense> <http://lexonomy.elex.is/elexis-oeaw-schranka#sense:25132_1> .
<http://lexonomy.elex.is/elexis-oeaw-schranka#sense:25132_1> <http://www.w3.org/2004/02/skos/core#definition> "den Hals abschneiden."@de .
<http://lexonomy.elex.is/elexis-oeaw-schranka#Abpaliieren_25133> <http://www.w3.org/1999/02/22-rdf-syntax-ns#type> <http://www.w3.org/ns/lemon/ontolex#LexicalEntry> .
<http://lexonomy.elex.is/elexis-oeaw-schranka#Abpaliieren_25133> <http://www.w3.org/2000/01/rdf-schema#label> "Abpaliieren"@de .
<http://lexonomy.elex.is/elexis-oeaw-schranka#Abpaliieren_25133> <http://www.w3.org/ns/lemon/ontolex#sense> <http://lexonomy.elex.is/elexis-oeaw-schranka#sense:25133_1> .
<http://lexonomy.elex.is/elexis-oeaw-schranka#sense:25133_1> <http://www.w3.org/2004/02/skos/core#definition> "mit der Zeche durchgehen."@de .
```

Figure 5: Example of dictionary export into OntoLex RDF format.

## 7. Standardisation

The ELEXIS project develops a standardised data model for digitally-born dictionaries as part of the OASIS LEXIDMA technical committee<sup>4</sup>. When the standardised format is published, Lexonomy will switch to the LEXIDMA data model as a default template for dictionaries. Keeping the complete configurability of custom user formats, of course.

In the meantime, Lexonomy supports TEI-Lex0 (Romary & Tasovac, 2018) and the OntoLex RDF format for ontologies (McCrae et al., 2017) as temporary formats. Lexonomy was updated to support both formats in API interfaces and to be integrated into automated lexicographic pipelines in the ELEXIS.

<sup>4</sup> [https://www.oasis-open.org/committees/tc\\_home.php?wg\\_abbrev=lexidma](https://www.oasis-open.org/committees/tc_home.php?wg_abbrev=lexidma)

## 8. Usage analysis

As of April 2021, over 2,700 users are working with Lexonomy. Altogether, they created more than 5,400 dictionaries containing over 34 million entries.

### 8.1 *OneClick Dictionary* dictionaries

Thanks to the connection with the Sketch Engine and its OneClick Dictionary tool, it is possible to create a new dictionary with the data from the corpus (e.g., headwords, examples). Utilising the OneClick Dictionary tool, users created 798 dictionaries in Lexonomy, which shows the popularity of automatic dictionary creation and post-editing. Most dictionaries cover a particular topic, e.g., terms from sports, medical science, or computer science. The most popular language with OneClick Dictionaries is English, followed by Czech, Italian and Latvian. Users created dictionaries in 30 different languages.

### 8.2 ELEXIS lexical resources

We have obtained 75 lexical resources from ELEXIS partners and observers (coming from 25 different institutions). The lexical resources range from different types of dictionaries, e.g., large general dictionaries, bilingual dictionaries, thesauri, specialised dictionaries (terminology, dialects), to lemma lists. Resources that were available in the XML format were directly uploaded to Lexonomy in their original format. Several resources were provided in different file formats, e.g., CSV or JSON. They were converted to the XML format before uploading to Lexonomy. Several dictionaries were provided in the PDF format, and these were converted to the XML format using the Elexifier tool. We list the largest resources (in terms of number of entries) in Table 1. Lexical resources provided by partners and observers are not publicly available, until licences are settled and exact access rights are specified. The Lexonomy application takes care of user accounts and access setting.

## 9. Conclusion

This paper summarises about two years of Lexonomy development. We introduced several features for a better user experience that attracted many new users to work with Lexonomy. Currently, over 2,700 users edit their dictionaries with Lexonomy, and we hope this number will grow even more. Other important updates include features that are integrating Lexonomy in various automated lexicographic pipelines. These integrations highlight the post-editing aspect of dictionary editing, and Lexonomy provides cutting-edge technologies even for small lexicographic teams or even one-person dictionary projects.

### 9.1 Future work

We are aware that the graphical user interface of Lexonomy is getting more cluttered with new features over time, and is also not suitable for work on mobile devices. On the developer side, currently used HTML templates are getting harder to maintain and extend. We decided to redesign and also refactor the user interface completely. The new

Lexical resource	Institution	Licence	Number of entries
Nova beseda frequency lexicon	ZRC SAZU Scientific Research Centre of Slovenian Academy of Sciences and Arts	CC BY 4.0	2,251,151
Svenska Akademiens Ordlista	Swedish Academy	open access license	984,823
Swedish Academy Dictionary	Swedish Academy	open access license	550,424
The Dictionary of Standard Estonian 2013	Institute of the Estonian Language	academic	425,766
Monier-Williams Sanskrit-English Dictionary	Cologne Center for Humanities	CC BY 3.0	398,412
Tezaurs Latvian	Institute of Mathematics and Computer Science, University of Latvia	CC BY-SA 4.0	320,869
The lemma list of the German dictionary "elexiko"	Leibniz Institute for the German Language	open access	275,756
Czech lemma lists	Institute of the Czech National Corpus	CC BY-SA 4.0	169,934
Dictionary of the Danish Language - ODS lemmas	The Society for Danish Language and Literature	restricted	163,012
Finnish dialect dictionary	Institute for the Languages of Finland	CC BY 4.0	161,148
Schweizerisehes Idiotikon	Schweizerisehes Idiotikon	CC BY-SA	160,254
Nords Ordbank - Bokmal	University of Bergen Library	CC-BY	153,939

Table 1: Selected lexical resources from ELEXIS partners, with more than 150,000 entries, sorted by the number of entries

user interface is currently in development, utilising the JavaScript component library Riot.js<sup>5</sup> and interface design framework Materialize<sup>6</sup>.

As we keep including more dictionaries with an increasing number of entries (tens of thousands of entries is getting more common) in Lexonomy, we are constantly monitoring the database performance. If we notice a decrease in speed with large amounts of data, we will evaluate other databases to select the best storage for big lexicographic data.

## 10. Acknowledgements

The research received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 731015. This work has been partly supported by the Ministry of Education of ČR within the LINDAT-CLARIAH-CZ project LM2018101.

## 11. References

- Kilgarriff, A., Baisa, V., Bušta, J., Jakubíček, M., Kovář, V., Michelfeit, J., Rychlý, P. & Suchomel, V. (2014). The Sketch Engine: ten years on. *Lexicography*, 1. URL <http://dx.doi.org/10.1007/s40607-014-0009-9>.
- Kilgarriff, A. & Jackson, H. (2013). Using corpora as data sources for dictionaries. *The Bloomsbury Companion to Lexicography*. London: Bloomsbury, pp. 77–96.
- Krek, S., Kosem, I., McCrae, J.P., Navigli, R., Pedersen, B.S., Tiberius, C. & Wissik, T. (2018). European lexicographic infrastructure (elexis). In *Proceedings of the XVIII EURALEX International Congress on Lexicography in Global Contexts*. pp. 881–892.
- McCrae, J.P., Bosque-Gil, J., Gracia, J., Buitelaar, P. & Cimiano, P. (2017). The Ontolex-Lemon model: development and applications. In *Proceedings of eLex 2017 conference*. pp. 19–21.
- McCrae, J.P. & Buitelaar, P. (2018). Linking datasets using semantic textual similarity. *Cybernetics and information technologies*, 18(1), pp. 109–123.
- McCrae, J.P., Tiberius, C., Khan, A.F., Kernerman, I., Declerck, T., Krek, S., Monachini, M. & Ahmadi, S. (2019). The ELEXIS interface for interoperable lexical resources. In *Proceedings of the sixth biennial conference on electronic lexicography (eLex)*. eLex 2019.
- Měchura, M.B. et al. (2017). Introducing Lexonomy: an open-source dictionary writing and publishing system. In *Electronic Lexicography in the 21st Century: Lexicography from Scratch. Proceedings of the eLex 2017 conference*. pp. 19–21.
- Romary, L. & Tasovac, T. (2018). TEI Lex-0: A target format for TEI-encoded dictionaries and lexical resources. In *TEI Conference and Members' Meeting*.

This work is licensed under the Creative Commons Attribution ShareAlike 4.0 International License.

<http://creativecommons.org/licenses/by-sa/4.0/>



<sup>5</sup> <https://riot.js.org/>

<sup>6</sup> <https://materializecss.com/>