**LexiCorp: Corpus Approach to Presentation of Lexicographic Data**

Vladimír Benko
Slovak Academy of Sciences, L. Štúr Institute of Linguistics
Panská 26, 811 01 Bratislava, Slovakia
E-mail: vladimir.benko@juls.savba.sk

Abstract

We present an experiment aimed at integrating XML-encoded dictionary data with corpus processing tools. Tokenized, lemmatized and PoS-tagged, the dictionary data can be processed by a traditional corpus manager such as NoSketch Engine (NoSkE), with the main benefit being the availability of ad-hoc full-text queries, as well as queries restricted to certain structure elements, without having to know too much about the internals of the respective XML encoding. Loaded with data from several Slovak dictionaries, the beta version of the dictionary portal (referred to as LexiCorp) is already used by our lexicographers.

We demonstrate the LexiCorp operation in the “Simple Query” mode and the use of “Zone” attribute in queries. However, having in mind that all NoSkE functionalities are available, we can say that users of LexiCorp can now receive a powerful working tool.

As NoSkE is an open-source system and implementation of LexiCorp requires just a minor modification of dictionary data and NoSkE’s CSS style(s), this approach is applicable to similar lexicographic projects as well. Though not intended to be a replacement of a fully-fledged Dictionary Writing System, it can be conveniently used to supplement functionalities that may be missing there, such as the use of regular expressions, statistics based on XML attributes, and queries related to morphological forms of search expressions.

**Keywords:** Dictionary writing system; corpus manager; full-text querying; NoSketch Engine

1. Introduction

Two types of software systems are typically employed in compilation of dictionary entries. Dictionary Writing Systems (DWSs), such as TLex¹, iLex² or Lexonomy³, are used to define the respective entry structures and to fill them with the necessary data. Corpus managers, e.g., CQPWeb⁴ or (No)Sketch Engine⁵,⁶, are needed to query corpora and to analyse, aggregate and process lexical evidence gathered out of them, especially if the corpora are really large. These two types of tools can cooperate to a certain extent to provide for partial automation of certain tasks, e.g., extracting suitable

---

¹ https://tshwanedje.com/tshwanelex/
² http://groupbanker.dk/generic-en/index.htm
³ https://www.lexonomy.eu/
⁴ http://cwb.sourceforge.net/cqpweb.php
⁵ https://nlp.fi.muni.cz/trac/noske
⁶ https://www.sketchengine.eu/
collocations or example sentences by means of the TickBox Lexicography⁷.

Our paper presents a different type of co-operation between dictionary data and a corpus manager, and describes an experiment in the framework of which we use corpus tools for the presentation of data of the Dictionary of Contemporary Slovak Language⁸ (DCSL, Jarošová & Benko, 2012) that is currently being compiled at our Institute.

2. The DCSL Project

Dictionary compilation is a rather time-consuming process. Producing a single-volume dictionary typically takes several years, and projects of multi-volume academic dictionaries may take even several decades to complete. This was also the case of the DCSL, whose preparatory phase was initiated already in mid-1990s, while the actual compilation of its first volume started in early 2000s. As of 2019, three DCSL volumes have been published (SSSJ1, 2016; SSSJ2, 2010; SSSJ3, 2016), two more volumes are currently in preparation, with the fourth volume being scheduled to be published in the end of the next year. The whole set is planned to consist of eight to nine volumes, which is most likely to occupy our lexicographic team for (at least) the next decade.

Partly due to historical reasons, our authors and editors do not work with the dictionary text in a “fully structured” format encoded in a generalized markup language, such as SGML or XML, and they instead use a light-weight markup language LLML (Benko, 2018). This is also one of the reasons why no “real” dictionary writing system (DWS) has been used yet for compilation of the DCSL.⁹

During the early “MS-DOS times” authors could prepare the text of the dictionary entries with any simple text editor, even with the built-in “F4 Editor” of Norton Commander ¹⁰. With the advent of MS Windows, the most convenient editing environment has been provided by the popular Notepad++ program¹¹ featuring user-definable syntax highlighting that could be easily adapted to our LLML syntax. Two sample entries as seen on the Notepad++ screen are shown in Figure 1.

---

⁷ https://www.sketchengine.eu/user-guide/user-manual/tickbox-lexicography/
⁸ http://www.juls.savba.sk/pub_sssj.html
⁹ The LLML approach has been used for all lexicographic projects carried out by our Institute since early 1990s, with the advantage being the high level of compatibility of all the lexicographic data, as well as the associated custom software tools.
¹⁰ https://en.wikipedia.org/wiki/Norton_Commander
¹¹ https://notepad-plus-plus.org/
It has been said that XML has not been used by the dictionary authors. It has been, however, used as an intermediate format during transformation of the dictionary text to the final printed and/or electronic form. The respective XML tags in this case represent typographical parameters, and can be easily mapped to typefaces, point sizes, colours, etc. Figure 2 shows an example of such XML code.

```xml
1 <en id="101_lo_wl_014960"> hword="lexikón">
2 <p class="main">b1<ch0><Skolexikón/Sk</bp0)</bp1> <i3>-nu/-na/ i3/ <tt0><pl. N</tt0> -ny | *m.| <gr.>
3 1; súhrnný zoznam slov z určitého odboru spracovaný
4 encyklopedicky, vykladový náučný slovník: 'biografický
5 l.; spoločenský l.' o etikete: 'l. slovenských dejín,
6 obcí; detský obrázkový l.; výstava hesla v leksikône;
7 vydávaný encyklopedie a leksikóny'; [prem.] '65-ročné
8 učiteľa pokladajú za živý leksikón.' [FN 1982]
9 vzdelaného, múdrego človeka
10 2) slovná zozba jazyka, lexika, ktorou disponujú
11 jeho používateľa, zásobovanie lexikálnych jednotiek:
12 'jednotky lexicónu'
13
14 <en id="101_lo_wl_014970"> hword="lexikónový" -vá -vé [*príd.]
15 0) vztahujúci sa na lexikon, náučný slovník; typický
16 pre lexikon: 'lexikónové diela; lexicónová definícia;
17 1. spôsob vykladu'
```

Figure 2: DCSL entries in “typographically motivated” XML notation.
3. Dictionary as a corpus

An XML-encoded dictionary is usually much more structured than a typical corpus. On the other hand, it can be treated as if it is a corpus. If processed by a standard tokenization and tagging pipeline for the respective language(s), it can be incorporated into a corpus manager without too many modifications needed.

The basic idea of our experiment is straightforward: as the procedures necessary to build and annotate (Slovak\textsuperscript{12}) corpora not only do exist but they have been fine-tuned already, we just need to find a way to “force” the corpus manager to display the dictionary structure in a format the lexicographers are accustomed to, i.e., structured by entries and highlighting the respective entry elements by means of typographical devices (such as point size, bold, italics, and colour).

3.1 Why NoSketch Engine

Our decision has been motivated by several factors. Firstly, as heavy users of the Sketch Engine (Kilgarriff et al., 2014), our lexicographers are also reasonably familiar with the environment of NoSketch Engine (NoSkE, Rychlý, 2007), and no additional training is expected. Secondly, the user interface provides for complex types of queries by means of the Corpus Query Language (CQL), yet it also offers “structure-agnostic” full-text querying in the Simple query mode. And lastly, the NoSkE client allows a simple way to customize the formatting of the output though mapping the respective user-defined XML structures into suitable CSS styles. Moreover, as NoSkE is available under the open-source licence, we will be able to share our solution with other lexicographic projects.

The customized version of NoSkE containing the processed data as installed at our dictionary portal is further referred to as LexiCorp.

3.2 Preparing the data

Any XML-encoded dictionary data can be easily incorporated into NoSkE, after being converted to a compatible “vertical” format and subsequently processed by a standard corpus-processing pipeline. This contains the following steps:

- Tokenization by the unitok\textsuperscript{13} (Michelfeit et al., 2014) tool using a custom parameter file (to take into consideration the dictionary-specific abbreviations and tokens starting and ending with hyphens used to indicate suffixes and prefixes in inflected

\textsuperscript{12} This applies, more or less, to any language with a morphosyntactic tagger available.

\textsuperscript{13} http://corpus.tools/wiki/Unitok
headword forms and elsewhere).

- Tagging by TreeTagger\(^{14}\) (Schmid, 1994) using a standard Slovak language model (Benko, 2016).
- Post-processing – fixing lemmatization and tagging issues for dictionary-specific out-of-vocabulary (OOV) tokens.
- Mapping native tags to a universal tagset\(^{15}\).
- Mapping the suitable corpus structure elements into <doc>, <p> and <s> structures used by default by the corpus manager (all other structures are preserved).
- Mapping dictionary structures into additional corpus attributes (to simplify certain types of queries).
- Indexing (“compilation”) by NoSkE.

### 3.3 Controlling the display

The standard NoSkE device for controlling the format of the richly structured corpora is the DISPLAYCLASS parameter that can be defined for each corpus structure contained in the corpus configuration file\(^{16}\). To make it operational, the appropriate CSS style has to be defined in the view.css file used by NoSkE. In a typical case, the respective dictionary XML structures have to be associated by a set of typographical parameters, such as typeface, point size and colour, which is fairly straightforward. Some CSS wizardry is needed only if some special effects (such as injections of newlines) are required.

### 4. First impressions

At the time of writing this paper (June 2019), the beta version of our LexiCorp installation contains data of all already published contemporary Slovak dictionaries produced by our Institute, as follows:

- Three volumes the Dictionary of Contemporary Slovak Language (SSSJ1, 2006; SSSJ2, 2010; SSSJ3, 2015)
- Live database of the Orthographic-Grammatical Dictionary (OGS, 2019)

\(^{14}\) [http://www.cis.uni-muenchen.de/~schmid/tools/TreeTagger/](http://www.cis.uni-muenchen.de/~schmid/tools/TreeTagger/)

\(^{15}\) [http://unesco.uniba.sk/aranea_about/aut.html](http://unesco.uniba.sk/aranea_about/aut.html)

\(^{16}\) [https://www.sketchengine.eu/corpus-configuration-file-all-features/](https://www.sketchengine.eu/corpus-configuration-file-all-features/)
Besides that, *LexiCorp* also contains data of two volumes of *DCSL* (SSSJ4, SSSJ5) that are currently being in preparation, as well as merged data of all dictionaries (less the dialectal ones). The *LexiCorp* home page\(^\text{17}\) is shown in Figure 3.

To demonstrate the basic functionality of the system, we will show some examples.

The easiest way to work with *LexiCorp* is to use the *Simple query* mode of *NoSkE* that is suitable for most “structure-agnostic” searches. For example, if we want to find all entries containing a certain phrase, we could do it like this (see Figure 4):

\(^{17}\) The *LexiCorp* portal containing data of the dictionaries currently being in preparation is not accessible to the general public, a *LexiCorp* demo site, however, containing the GNU Collaborative International Dictionary of English (*GCIDE*, http://gcide.gnu.org.ua/) is already available at: http://lexicorp.juls.savba.sk/guest.
Figure 5: Majúci velký (“having large”)

We can notice here several things. The “Short reference” on the left part of the display contains the Id of the dictionary (“1c” meaning the first volume of SSSJ), and the respective headword. The display mode was set to “Sentence”, which has been mapped to one sense in this particular dictionary.

As the dictionary text has been lemmatized (and also morphosyntactically tagged), LexiCorp can find the respective expression in all morphological forms – this is something a traditional DWS is typically not capable of.

The search expression is a phrase typically contained in dictionary definitions, and is hard to find elsewhere – we, therefore, do not have to bother about the dictionary structure while querying.

The entry is structured by means of typography, leaving NoSkE to highlight search expression by the default red colour.

Similarly, it is quite easy to make a query based on an abbreviation (See Figure 6).
Figure 6: Port. (Words of Portuguese origin)

Or, just a combination of metalanguage elements (see Figure 7).

Figure 7: Pl. N -ci (Words with a particular form in the plural nominative case)
5. The second round

Though users could use the CLQ mode of NoSkE to look up expressions and strings within the various dictionary structure fields, such as headword, definition, example, etc., this would not be a good solution in our situation as our lexicographers are rather reluctant to learn anything “too abstract”.

We therefore decided to employ the part-of-speech (PoS) filter of NoSkE that can be set for Lemma and Word form queries. (See Figure 8).

![Figure 8: PoS filter](image)

The PoS filter is based on mapping morphological tags provided by tagger into “readable” names of PoS defined in the corpus configuration file.

As NoSkE “does not care” about the actual values assigned to PoS, this functionality can be used to filter any attribute attached to the respective token(s), if appropriate mappings are supplied. In our case, the mappings were based on entry structure elements, such as headword, definition, example, etc.

So that the user would not be confused, we changed the “PoS” string in the menu to “Zone”, which was, in fact, the only modification of NoSkE source code necessary (see Figure 9).

![Figure 9: Query within the heslo (“headword”) zone](image)

Using this functionality, the user does not need to know the names of the respective XML elements that encode the particular “zones”, which makes the system more...
accessible also for linguists not directly involved in the dictionary compilation.

In our example, the regex functionality of NoSkE is used to look up for all headwords related to lexicography in all dictionaries stored in LexiCorp, and the “1st hit in doc” filter is applied to get rid of multiple occurrences of entries caused by run-on headwords. The result is shown in Figure 10.

6. “Bells and whistles”

The beta version of LexiCorp turned to be a success and was “warmly welcomed”, not only by the lexicographic team members but by also by the other researchers at our Institute. This was probably the reason why no large-scale modification has been attempted since. Here are some small points to mention.

6.1 Merged dictionary data

After the unification of structures of our dictionaries, we managed to merge all data into one resource that can be conveniently looked up with a single query as shown in the previous chapter. Due to the unified format used to represent our dictionaries (Benko, op. cit.), this operation was relatively easy to perform. We must admit, however, that this needs not be the case if new dictionaries with more richly structured entries are to be incorporated into LexiCorp.
6.2 Typography

The graphical representation is very important when dictionary data are displayed on a computer screen. We made a series of experiments aimed at improving the legibility of the output. As a consequence, we decided to change the default sans-serif typeface used by NoSkE for displaying the concordances (i.e., the dictionary entries) to a serif one that better distinguishes between Roman and italicized text within the entries. As all our users work on Microsoft Windows machines, we opted for a standard Windows Georgia\textsuperscript{18} font that is known to have been designed with screen readability in mind.

Paper versions of our dictionaries use several special characters (custom created by a font editor) to introduce special sections of entry, such as lexicalized expressions, idioms, run-ons, etc. Some of these characters do not even have a similarly looking Unicode equivalent. To make the problem of displaying these characters easier to solve, we decided to substitute them for different ones (sometimes not even resembling the original glyphs) selected from the Font Awesome\textsuperscript{19} icon collection, that is used internally by NoSkE and therefore already installed in the system.

The text colours of the respective dictionary zones were chosen to be compatible with those used within the dictionary production environment (Benko, 2018), i.e., so that the lexicographers would see them as familiar.

A LexiCorp logo and a favicon have also been designed, so that the Portal had a unified “look”.

6.3 Dictionary names

Similarly to naming convention within the Aranea web corpora project (Benko, 2014), the respective dictionaries were assigned “language neutral” (Latin) names\textsuperscript{20}, as well as two-character Ids that are displayed along with the headwords in the “short reference” zone at the left side of the output screen.

7. Conclusion and further work

The experiment presented in this work proved the feasibility of our approach. The server component of NoSkE proved to be more than adequate for the task. The problem of the client is that is “too good”, i.e., contains too many features not necessary for typical dictionary look-ups that may confuse (especially inexperienced) users. It could

\textsuperscript{18} https://en.wikipedia.org/wiki/Georgia_(typeface)

\textsuperscript{19} https://fontawesome.com/

\textsuperscript{20} It may be interesting to note that in the territory of today’s Slovakia Latin was used as an official language until the middle of the 19th century.
be, however, a good start for building a specialized client – this is, however, beyond our capacity. We are willing, however, to provide our know-how and data structures to anyone interested.

Readers may be wondering what could be the advantages of using LexiCorp instead of a full-fledged DWS. We are, however, not arguing in favour of using it instead, but rather in parallel. We hope that the main advantages have been addressed in the previous text.

As the compilation of LexiCorp out of the source dictionary data at our site is now fully automated and lasts less than 20 minutes, it can be performed regularly, theoretically even on the daily basis so that the lexicographers can work with fresh data every day. At the present stage, however, we have found that once a week is fully sufficient.

8. Acknowledgement

This work has been, in part, funded by the VEGA Grant Agency, Project No. 2/0017/17.

9. References


**Dictionaries:**


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

[http://creativecommons.org/licenses/by-sa/4.0/](http://creativecommons.org/licenses/by-sa/4.0/)