Michal Měchura

Contributions to e-lexicography

Ph.D. thesis proposal

teze dizertační práce

Fakulta informatiky Masarykovy univerzity

V Brně dne

podpis školitele

0. Introduction

0.1. What is lexicography

This thesis will be about dictionaries created for human users (readers, consumers). Dictionaries for humans are different from machine-oriented language resources as used in natural language processing (NLP). Although there is some overlap between these two types, my thesis will be only about the first type and I will use the term *lexicography* to mean *human-oriented lexicography* only.

When we constrain the definition of lexicography to target humans exclusively, it becomes natural to think of lexicography as a subdiscipline of *information science* (a term more or less synonymous with *library science* and *reference science*). Information science is the activity of taking knowledge from a given field (physics, economics, ...) and then organizing and presenting this knowledge in artefacts such as libraries and encyclopedias, in ways which are likely to satisfy the *information needs* of human users. The parallels to what lexicographers do are obvious: lexicography is the activity of taking knowledge about words (which comes from a related discipline called *lexicology*) and then organizing and presenting this knowledge in artefacts called *dictionaries* which are built to serve the *information needs* of human users. Throughout my thesis, I am attempting to bring digital innovations into lexicography which would result in a better satisfaction of the information needs of human dictionary users.

0.2. The history and future of digitization in lexicography

Following Atkins and Rundell 2008 (p. 3) there are three stages in the dictionary-making process where computer software comes in: (1) as corpus query systems for discovering lexical knowledge in corpora, (2) as dictionary writing systems where lexical knowledge is encoded into a form suitable for presentation to human readers and (3) as websites, apps etc. which deliver the dictionary onto the end-user's screen. Together these three areas constitute the discipline known as *e-lexicography*.

(1) Most innovation in e-lexicography so far has happened in corpus query systems: so much, in fact, that corpus-driven methods have redefined dictionaries from intuition-based prescriptions to evidence-based descriptions.

(2) The area where the least amount of innovation has happened until now is the middle part, dictionary writing. Even though dictionary writing has become completely computerized in the last few decades, the structure of dictionaries we write today has not changed since pre-computer times. I will show in my thesis that the way in which dictionary writing has been digitized, through XML encoding in dictionary writing systems (DWS), has been only superficial and rather unhelpful. Tasks which were difficult in pre-computer times are still difficult today, such as the reversal of bilingual dictionaries or the placement of multiword phrases. The digital medium has the potential

to make these tasks easy but, to avail of that potential, we must re-engineer the internal structure of dictionaries.

(3) At the end of the pipeline, in dictionary publishing, websites and other electronic media have now almost completely replaced the printed dictionary. Importantly, screen-based dictionary publishing is becoming divorced from the original print medium. We are seeing the emergence of well-designed dictionary websites with an excellent user experience which no longer confine themselves to merely imitating on computer screens what had previously been done on paper, and which serve their users' information needs better than printed dictionaries ever did. This is a welcome development but, again, we have not exhausted yet the full potential offered by the digital networked environment in which dictionaries are published nowadays.

My thesis will have little to contribute under (1) but will make several proposals for innovation under (2) and (3).

0.3. Thesis topics

My thesis breaks down into four semi-independent topics, each anchored at a different point along the dictionary-making pipeline outlined above.

Topic 1: Structural markup in dictionary encoding. Dictionary entries are usually encoded in XML and, increasingly in recent years, in JSON. Both these languages encourage the use of excessive structural markup which distracts human editors from the lexicographic facts they seek to encode. In my thesis I will propose a different formalism called *name-value hierarchy* in which dictionary entries can be encoded with more human-readability (and writability).

Topic 2: Graph-augmented trees. Dictionary entries are usually encoded as tree structures (regardless of whether the encoding formalism is XML, JSON or a name-value hierarchy). But tree structures make certain lexicographic tasks unnecessarily difficult, such as the re-use of information in multiple places in the same dictionary, or reversing a bilingual dictionary. In my thesis I am proposing a new data structure called *graph-augmented trees* which makes these tasks easier.

Topic 3: Lexicographic semantics. When software tools (such as dictionary writing systems or internet search engines) process dictionary entries, they are mostly unaware of what the different text segments in the entry mean and what role they play in the entry: whether something is a headword, a definition, a sense etc. The boundaries of the segments are clear from the markup but their *lexicographic semantics* is not. In my thesis I am proposing an *inventory of lexicographic information types* which can be used to formally specify the lexicographic semantics of various dictionary entry segments.

Topic 4: Dictionary aggregation. Dictionaries are usually published as websites. Each dictionary website is different and this makes it difficult to build aggregation tools such as metasearch engines and dictionary portals where users could consult multiple dictionaries at the same time, on a single screen. The main obstacle is that web-based dictionaries are understandable for humans but not for machines. In my thesis I am proposing several dictionary-specific extensions to existing web standards such as *OpenSearch* and *Schema.org* (based on the *inventory of lexicographic information types* from Topic 3) which will give dictionary publishers a lightweight method for making their dictionaries (and/or metadata about them) available on the web in machine-readable formats.

Each topic is explained in a separate chapter in the remainder of this proposal.

0.4. Relation to other work

My work on this thesis will happen within ELEXIS, an EU-funded project with an ambition to develop next-generation infrastructure for lexicography in Europe. I am involved in ELEXIS through Lexical Computing, a Masaryk University campus company which is sponsoring my research.

One of the cornerstones of ELEXIS is Lexonomy, an open-source dictionary writing and publishing system which I created in 2016. Since its inception Lexonomy has become a community-driven project and is currently being both used and developed by several dictionary publishing institutions in Europe and elsewhere (including Lexical Computing). Developing Lexonomy further is one of the goals of the ELEXIS project. Many of the techniques and patterns proposed in my thesis will be (or already are) implemented in Lexonomy, including:

— Lexonomy will use *name-value hierarchies* (Topic 1) in its editing interface.

— The tree structures stored as name-value hierarchies in Lexonomy will be (and, partially, already are) *graph-augmented trees* (Topic 2). This will support various Lexonomy features such as subentry sharing and bilingual reversal.

— Lexonomy will use my *inventory of lexicographic information types* (Topic 3) to 'understand' the lexicographic semantics of dictionary entries. This will support several features throughout Lexonomy, from simple ones (such as knowing which part of the entry is the headword) to complex ones (such as a configuration wizard to quickly set up a new dictionary project).

In addition to Lexonomy I am also the technical architect behind the European Dictionary Portal, a website developed within the ENeL project, a precursor to ELEXIS. Developing a dictionary portal is another goal of ELEXIS. The aggregation mechanisms I will be proposing in Topic 4 will be implemented in a re-engineered version of the European Dictionary Portal as well as several dictionary websites which feed the portal (including dictionaries published through Lexonomy).

0.5. Author's publications and presentations

A general introduction to Lexonomy:

— Michal Měchura (2017) *Introducing Lexonomy: an open-source dictionary writing and publishing system.* Proceedings of eLex 2017 conference, Leiden, the Netherlands.
`https://michmech.github.io/pdf/elex2017.pdf`

Descriptions of how Lexonomy fits into the modern lexicographic process:

— Miloš Jakubíček, Michal Měchura, Vojtěch Kovář, Pavel Rychlý (2018) *Practical Post-Editing Lexicography with Lexonomy and Sketch Engine.* Proceedings of XVIII EURALEX International Congress, Ljubljana, Slovenia.

— Miloš Jakubíček, Vojtěch Kovář, Michal Měchura, Pavel Rychlý (2017) *One-Click Dictionary.* Presentation at eLex 2017 conference, Leiden, the Netherlands.

A summary of my work on the European Dictionary Portal:

— Michal Měchura (2017) *How (not) to build a European Dictionary Portal.* Final Conference of the European Network of e-Lexicography, Leiden, the Netherlands.
`https://www.youtube.com/watch?v=ORrGelo9ytU`

1. Structural markup in dictionary encoding

1.1. State of the art

Dictionary encoding is the activity of taking an inventory of lexicographic object types such as headword, part-of-speech label, sense and translation, and expressing them formally in a data serialization language such as XML.

1.1.1. The problem of excessively complex markup

The use of XML for dictionary encoding often leads to excessively complex markup, with multi-layered embedding of elements inside other elements inside yet more elements. Code sample 1 shows how a pair of translations would typically be encoded in a bilingual dictionary.[1]

Code sample 1

```
<translations>
  <translationContainer>
    <translation>leasú</translation>
    <pos>n-masc</pos>
  </translationContainer>
  <translationContainer>
    <translation>athchóiriú</translation>
    <pos>n-masc</pos>
    <usage>formal</usage>
  </translationContainer>
</translations>
```

The only XML elements here that contain actual human-readable information are `<translation>` (the translation's wording), `<pos>` (its part of speech) and `<usage>` (its usage label). The remaining XML elements are purely structural, used for grouping other elements together. Arguably, their presence here distracts a human XML reader (and even more so, a human XML writer) from lexicographic information which is otherwise simple and could be expressed more economically in some other (not yet existent) serialization language such as the pseudo-code in Code sample 2.

[1] The natural language in code samples in this chapter is Irish, and the XML code is adapted from the New English–Irish Dictionary (Ó Mianáin and Convery 2014) when not otherwise stated.

Code sample 2

```
translation: leasú
  pos: n-masc
translation athchóiriú
  pos: n-masc
  usage: formal
```

The distracting presence of purely structural elements in lexicographic XML is often acknowledged as an inconvenience in e-lexicographic circles informally but, to my knowledge, no serious attempts have been made yet to analyze or solve it.

1.1.2. Patterns of purely structural markup in lexicographic XML

We can define *purely structural markup* as such XML elements which contain no text nodes as their direct children: all their child nodes are other XML elements. Purely structural elements tend to be called *groups*, *containers* or *blocks* in the entry schemas of various dictionaries. For example, the entry schema for the DANTE project (Atkins, Kilgarriff and Rundell 2010) consists of elements such as `<CollocGp>` (collocate group) as a wrapper for a sequence of one or more collocates, `<CollocCont>` (collocate container) as a wrapper for a single collocate along with additional information about it (usage labels, example sentences, translations etc.) and finally `<COLLOC>` as a wrapper for the actual collocate (a text node). The first two of these three element types are purely structural.

Broadly speaking, we tend to find two patterns of purely structural markup in lexicographic XML.

List pattern. The first kind is a pattern in which a parent element wraps a sequence of child elements which are all of the same type, such as `<CollocGp>` for a series of collocates in Dante, or `<translations>` for a series of translations in Code sample 1. They are almost always unnecessary in the sense that they convey no useful information. They are there because the designer of the entry schema probably thought it 'logical' to group elements of the same type under a common parent element. But the usefulness of this grouping is debatable: the group thus created does not seem to represent any lexicographic fact which a lexicographer might want to communicate to the dictionary's end-users. Unnecessary grouping of this kind can be found in XML outside lexicography too and tends to be advised against in XML styleguides (eg. Ogbuji 2004).

Headed pattern. The second kind is a pattern in which a parent element wraps child elements of different types, one of which can be considered the *head* and the others can be seen as providing additional information about the head. An example is `<translationContainer>` in Code sample 1 which can be said to be headed by `<translation>`, while the other children `<pos>` and `<usage>` provide additional information about the head. In DANTE, a similar example is `<CollocCont>`

which is headed by `<COLLOC>` (the actual collocate) while other child elements of `<CollocCont>` provide additional information about the head (usage labels, example sentences, translations etc.).

Unlike the list pattern, the headed pattern cannot be explained away as a bad practice. Its purpose is to encode a lexicographic fact which the lexicographer wants to communicate to the end-user: for example, which `<pos>` element modifies which `<translation>` element. The purely structural `<translationContainer>` element is a tool for encoding that fact. The question to ask now is whether there are other ways of encoding that fact: whether the headed pattern can be encoded in XML without recourse to purely structural markup.

1.1.3. Encoding the headed pattern in XML

One obvious suggestion is to encode non-head objects as XML attributes of the head, so there is no need for a purely structural parent, as in Code sample 3.

Code sample 3

```
<translation pos="n-masc" usage="formal">leasú</translation>
```

But XML attributes come with two inconvenient limitations: an element cannot have two or more attributes of the same name, and the value of an attribute must be plain text (ie. it cannot have child nodes or attributes of its own). This means that examples like those in Code sample 4 and Code sample 5 cannot be re-encoded through attributes:

Code sample 4

```
<translationContainer>
  <translation>leasú</translation>
  <pos>n-masc</pos>
  <pos>verbal-noun</pos>
</translationContainer>
```

Code sample 5

```
<translationContainer>
  <translation>athchóiriú</translation>
  <usageContainer>
    <modifier>mostly</modifier>
    <usage>formal</usage>
  </usageContainer>
</translationContainer>
```

Another possible suggestion is to encode the non-head elements as children of the head, thus again eliminating the need for a purely structural parent, as in Code sample 6.

Code sample 6

```
<translation>
  leasú
  <pos>n-masc</pos>
</translation>
```

This has the disadvantage of removing the formal and explicit distinction between text which constitutes the head and text which does not. A typical XML parser when asked to give the value of `<translation>` would return the string `"leasú n-masc"`. In a simple example like this one, we could work around that by declaring that only the first text node constitutes the head. But this becomes more cumbersome if the head contains its own child elements, like it does in Code sample 7.

Code sample 7

```
<example>
  d'air an pobal <b>leasú</b> toghchánach
  <translation>the public demanded electoral reform</translation>
</example>
```

Here it becomes impossible, without an understanding of the entry schema, to tell where the head (the example sentence) ends and where non-head content (its translation) begins.

The interim conclusion is that, in the general case, it is not possible to encode the headed pattern in XML without purely structural markup. Additionally, the problem cannot be solved simply by migrating to some other well-known serialization language such as JSON or YAML because these share one crucial property with XML: they possess no formal means of encoding *headedness* other than by purely structural markup.

1.2. Aim of the thesis

In my thesis I will discuss the challenge of purely structural lexicographic markup in depth. I will probably reach the conclusion that some instances of purely structural markup are unavoidable and that the problem is unsolvable, as long as one insists on using XML or another well-known serialization language such as JSON or YAML. What XML, JSON and YAML have in common is that they take no account of the inherent *headedness* of many lexicographic information objects.

There are two ways in which one could remove the inconvenience caused by purely structural markup.

Option 1. Invent a new serialization language which avoids purely structural markup and respects headedness. In such a language, a dictionary entry would be a *hierarchical list of name-and-value pairs* or, more concisely, a *name-value hierarchy*. The language would read similarly to the pseudo-code in Code sample 2, would be more easily human-readable and human-writable than XML or JSON, and would be superficially similar to YAML (but would not be a subset of it).

Option 2. Build editing tools which hide the purely structural markup from the human editor while still keeping the underlying data in XML (or JSON). The human editor would work with a representation which looks like the pseudo-code in Code sample 2 while, behind the scenes, the tool would take care of converting the data from and into XML (including purely structural markup).

In my thesis I will investigate which of the two options is more workable. If it turns out that Option 1 is more workable, the language of name-value hierarchies I will thus create will be similar in principle to a lexicographic lightweight markup language *à la* Benko (2018). If Option 2 turns out to be the more practical option then the new language will serve only as a vehicle for a non-persistent surface representation of XML in the fashion of *Invisible XML*[2] (Pemberton 2013).

1.3. Results achieved so far

An informal description of name-value hierarchies already exists. No decision has been made yet whether it will be independent of XML (Option 1) or whether it will be a simplified surface representation for XML (Option 2).

1.4. Results to be achieved yet

I will produce a formal specification of name-value hierarchies and program a reference implementation (a parser, a serializer and an object model), as well as as user interface for editing dictionary entries.

1.5. Author's publications and presentations

None yet. I am preparing a paper (together with colleagues in Lexical Computing) which will explain the concepts alluded to here, such as *headedness*, and present a detailed argument in favour of name-value hierarchies.

---

[2] Invisible XML is "a method for treating non-XML documents as if they were XML, enabling authors to write documents and data in a format they prefer while providing XML for processes [behind the scenes]" (Pemberton 2013).

2. Graph-augmented trees

2.1. State of the art

In lexicography, a dictionary entry is typically encoded as a tree: a hierarchical data structure of parent-child relations where every element has at most one parent.[3] This choice of data structure makes some aspects of the lexicographer's work unnecessarily difficult, such as deciding where to place multiword items or reversing a bilingual dictionary.

2.1.1. Placement of multiword items

A perennial problem in lexicography is deciding on the placement of multiword items (Bogaards 1990): should a phraseological unit such as *third time lucky* be located inside the entry for *third, time* or *lucky?* In many such cases the best imaginable answer is *under all of them*. But such a suggestion is difficult to accommodate in the classical model of dictionary entries as a tree structures. The only way to include a phraseological unit in more than one entry is to duplicate it, but this is an inelegant solution. Most importantly, it opens up the potential for inconsistency: if a lexicographer makes a change to the subentry *third time lucky* under *third,* there is no automatic way to propagate the change to the other copies under *time* and *lucky.*

A popular method to deal with this in born-digital dictionaries is to treat multiword phrasemes as independent entries, in effect promoting them to the same level as single-word headwords. This approach 'solves' the problem of multiword item placement by deciding not to place them anywhere, and that is also its drawback: it strips the lexicographer of the ability to include a multiword item like *third time lucky* in a specific sense of a single-word entry, for example in a specific sense of *time*. Instead, it delegates the placement question to the search algorithm, hoping that *third time lucky* will indeed appear somewhere on the end-user's screen when the he or she has looked up *time*. This is far from ideal: the job which an item like *third time lucky* does in a dictionary is not just that of a phraseme which users might look up independently. It is (or can be) simultaneously an illustrative example of specific senses of the words it is composed of. This means that the desire to include it in a specific location inside one or more specific entries is lexicographically well-motivated and the 'treat-multiwords-as-headwords' method is only a workaround. What is needed is a method for including a single multiword item in several locations inside several entries, but without having to keep multiple copies of them in multiple locations.

---

[3] It would be tempting to assume that the modelling of dictionary entries as tree structures was an innovation introduced into lexicography by the arrival of XML. That is not true. The 'imagining' of dictionary entries as theoretical tree structures predates even the invention of XML. A summary of this thinking from pre-computarization times can be found in Wiegand (1989). I am grateful to David Lindemann for this insight.

2.1.2. Bilingual dictionary reversal

Another well-known problem in lexicography is reversing a bilingual dictionary (Maks 2007). Once we have written a bilingual dictionary from language X to language Y, it is far from trivial to convert it into a dictionary that goes in the opposite direction, from language Y to language X. There are points of indeterminacy which prevent us from doing it completely automatically. More importantly, the process is a one-way street: once we have reversed the dictionary, we have lost the connection between the source and the target: each entry in each dictionary is its own tree structure with no explicit links between them. If and when the source dictionary changes, the reversed dictionary has potentially become outdated as there is no automated way to propagate changes from one into the other or to notify a human editor that changes have occurred. A more attractive proposition would be to encode pairs of bilingual dictionaries in a structure that keeps them synchronized, so that every element in every entry in the reversed dictionary 'knows' which element in which entry in the original dictionary it came from, and can react to changes. This calls for a graph-based data structure where each element can have relations with other things besides its hierarchical parent.

2.1.3. Are graphs the answer?

These two problems are well known in lexicography and have not been solved by digitization (ie. by moving dictionaries from paper to computers). They can be understood as an inconvenient consequence of the tree-like data structure dictionaries are encoded in.

It is possible to make these lexicographic tasks easier by re-engineering dictionaries as *graphs* rather than trees. Many NLP-oriented lexical resources are indeed organized as graphs, but this is rare in human-oriented dictionaries. A typical example is WordNet (Fellbaum 1998) and other semantic networks which, in effect, are models of the mental lexicon. These seem like a promising source of inspiration. Instead of writing a tree-structured dictionary, one could build a graph-based model of the mental lexicon and then *derive* dictionaries from it, automatically and on demand. The conventional tree-structured entry would become a non-persistent output format, one of many possible 'views' of the graph, while problems such as multi-word item placement and dictionary reversal would disappear. In practice, however, all attempts to build a human-oriented dictionary in this way have so far remained experimental (eg. Polguère 2004).

Lately, some dictionary producers have become inspired by the Semantic Web and started experimenting with re-encoding dictionaries as RDF graphs (eg. Aguado-de-Cea 2016, Klimek and Brümmer 2015). This is a more realistic attempt at innovation because, unlike semantic networks *à la* WordNet, it does not attempt to model the mental lexicon. Instead, it merely captures the same information dictionaries already have in trees and encodes it in a graph. In an RDF graph, dictionary entries can be augmented with various relations which 'break out' of the tree paradigm, for example

sense-to-sense links between synonyms. The relations envisaged above, such as many-to-many relations between multiword phrasemes and word senses, could be accommodated in an RDF graph easily. But, unfortunately, all RDF encodings of human-oriented dictionaries have so far been automatic conversions from pre-existing tree-structured XML and do not take advantage of these possibilities.

2.2. Aim of the thesis

In my thesis I will discuss the relative advantages and disadvantages of trees and graphs as data structures for encoding human-oriented dictionaries. I will come to the conclusion that the main disadvantage of graphs (and probably also the reason for their lack of adoption in the industry) is that they are not as easily human-readable as trees, not to mention human-writeable. Trees can be visualized neatly as two-dimensional objects, while graphs often can't. Trees are easy for humans to grasp mentally, while graphs are more difficult to 'take in'. For this reason, it is unlikely that lexicographers will switch to authoring graph-based dictionaries directly any time soon. The problem then is that, while graphs are the more adequate structure for dictionaries, trees are more 'lexicographer-friendly'.

I will propose a conservative compromise called *graph-augmented trees* in which existing tree structures become augmented with specific types of inter-entry relations designed to solve specific problems. In that model, dictionaries will continue to be written in conventional tree structures – or so they will appear to the lexicographers. Behind the scenes, the dictionary writing system will keep track of any relations that 'break out' of the tree and present them to the lexicographer as annotations beside the tree: for example, to allow the sharing of phraseological subentries between entries. Additionally, I will provide a dictionary-writing user interface which allows lexicographers to work with dictionary entries in the familiar tree format while only forcing them to 'think outside the tree' when necessary.

2.3. Results achieved so far

Graph-augmented trees have already been proposed and thoroughly argued for in a conference paper (Měchura 2016, see below). Lexonomy currently implements the graph-augmented tree model as a solution to the problem of multiword item placement (called *shareable subentries* in Lexonomy).

2.4. Results to be achieved yet

I will implement graph-augmented trees as Lexonomy's solution for bilingual dictionary reversal (more generally, for mapping a pair of dictionaries onto each other and keeping them synchronized semi-automatically), as envisaged in the original conference paper (Měchura 2016).

2.5. Author's publications and presentations

This paper describes the subentry sharing feature in Lexonomy:

— Michal Měchura (2018) *Shareable Subentries in Lexonomy as a Solution to the Problem of Multiword Item Placement.* Proceedings of XVIII EURALEX International Congress, Ljubljana, Slovenia.
`https://michmech.github.io/pdf/euralex2018.pdf`

This paper presents a detailed argument in favour of graph-augmented trees, and discusses how graph-augmented trees could be used to solve the multiword placement problem and for bilingual dictionary reversal:

— Michal Měchura (2016) *Data Structures in Lexicography: from Trees to Graphs.* Proceedings of Recent Advances in Slavonic Natural Language Processing, Masaryk University, Brno.
`https://michmech.github.io/pdf/raslan2016.pdf`

3. Lexicographic semantics

3.1. State of the art

3.1.1. Lexicographic semantics in dictionary writing systems

Modern dictionary writing systems (such as iLex, IDM DPS, TLex and the current incarnation of Lexonomy) are basically schema-aware XML editors. They are mostly unaware of what the different text segments in the entry mean and what role they play in the entry: whether something is a headword, a definition, a sense etc. The boundaries of the segments are clear from the markup but their *lexicographic semantics* is not.

XML elements with the same or similar lexicographic semantics can have different names in different dictionaries. For example, a headword can be called `<headword>` or `<hwd>` or `<lemma>` or indeed anything. There is now way to determine the lexicographic semantics of an element just from the names the authors had decided to give them.

The consequence is that some features, such as cross-referencing and reverse indexing, are unnecessarily complex to implement when developing a dictionary writing system, and unnecessarily complex to configure in the dictionary writing system for each individual dictionary. For example, if we know that an XML element is a container for example sentences, this may have the following implications for features in our dictionary writing system:

— The XML element's content should be fulltext-indexed by the dictionary writing system and made available for fulltext search.

— This is the XML element which should be inserted into the entry when the dictionary writing system is pulling example sentences in from a corpus.

— The dictionary writing system should treat the XML element as a potentially shareable subentry.

In a typical dictionary writing system (including Lexonomy) each such feature needs to be configured individually because the software does not understand that the XML element is an *example sentence:* all it knows that it is an XML element. The only (partial) exception is the dictionary writing system TLex which has a notion of 'special element types' which trigger the software to provide specific functionality such as cross-referencing (Joffe and de Schryver 2018, p. 60).

3.1.2. Lexicographic semantics on dictionary websites

When dictionaries are published online as websites, the entries are seen by search engines and web crawlers merely as sequences of HTML elements whose lexicographic 'meaning' is opaque: it is not clear where the headword is, where one sense ends and another begins, and so on. For example, Code

sample 1 shows the (simplified) HTML source code of the entry for *passable* on the website of Longman Dictionary of Contemporary English.

Code sample 1

```
<span class="ldoceEntry Entry">
  <span class="Head">
    <span class="HWD">passable</span>
    <span class="HYPHENATION">pass·a·ble</span>
    <span class="POS">adjective</span>
  </span>
  <span class="Sense" id="passable__1">
    <span class="sensenum span">1</span>
    <span class="REGISTERLAB">formal</span>
    <span class="DEF">fairly good, but not excellent</span>
    <span class="EXAMPLE">The food was excellent and the wine was passable.</span>
  </span>
  <span class="Sense" id="passable__2">
    <span class="sensenum span">2</span>
    <span class="DEF">
      a road or river that is passable is not blocked, so you can travel along
      it or across it
    </span>
  </span>
</span>
```

The class names such as `DEF` and `EXAMPLE` are specific to this website and cannot be used as a dictionary-independent indication of their HTML elements' lexicographic semantics. A web crawler visiting this website will not understand where the definitions and examples are. The consequence is that it is difficult to build dictionary aggregation tools such as dictionary portals (about which see Topic 4 next).

3.2. Aim of the thesis

In my thesis I will propose a formal *inventory of lexicographic information types* (such as headword, sense, example sentence, collocation, ...) which can be used to formally specify the lexicographic semantics of various dictionary entry segments. The inventory will be based on an analysis of a broad range of existing digital dictionaries (both born-digitial and retrodigitized) and entry schemas.

The inventory will not be 'just another' lexicographic data format like LMF or the TEI Guidelines Dictionary Chapter. It will not be an entry schema in which dictionary entries could be encoded or into which they could be converted. Rather, it will be a schema-agnostic inventory of types to be used as stand-off annotation for dictionary entries (and for dictionary entry schemas) in any format.

In a dictionary writing system such as Lexonomy, the inventory can (and, in Lexonomy, will) be used to simplify dictionary configuration. The dictionary administrator will specify which elements in the entry schema correspond to which information types in the inventory, and this will trigger the software to provide relevant functionality. For example, if the software 'knows' which entry segments are example sentences, it will automatically treat them accordingly, for example by full-text indexing them, by making them searchable, and by offering automatic corpus lookup to find more examples – there will be no need to configure these features individually.

On dictionary websites the inventory can be used to mark up the lexicographic semantics of entry segments as *HTML Microdata* embedded in HTML, in order to make it easier for web crawlers and similar tools to ingest lexicographic content from the web. More about this in Topic 4.

3.3. Results achieved so far

None yet. The need for a formalized lexicographic semantics has emerged from my work on Lexonomy and on the European Dictionary Portal.

3.4. Results to be achieved yet

I will publish the inventory as an open specification on the web, and explain the rationale for it in a peer-reviewed publication. The inventory will be implemented in Lexonomy as a tool for dictionary configuration, and will be used in my Topic 4 as a *HTML Microdata* vocabulary for dictionary websites.

3.5. Author's publications and presentations

I presented an initial outline of my vision for a formalized lexicographic semantics in this talk :

— Michal Měchura (2019) *The future of dictionary editing.* Invited talk at Lexicom, a training event organized by Lexical Computing, Mikulov, Czech Republic.

4. Dictionary aggregation

4.1. State of the art

When dictionary publishing started migrating from paper to screens in the early 2000s, it was a large innovation: one could now search a dictionary quickly and accurately in one or two seconds instead of having to leaf through the pages of a book. Today, this evolutionary step is in the past and online dictionaries have become the new normal. It is now time to ask: what is the next step going to be in the evolution of human–dictionary interaction?

4.1.1. The next step: aggregation

It seems reasonable to assume that the next step is going to be some form of *aggregation:* people will increasingly be demanding to access multiple dictionaries simultaneously, from one place. A similar trend already exists in other branches of information science including libraries (in the form of country-specific or city-specific library portals (like `cistbrno.cz` which meta-searches all libraries in the city of Brno or `knihovny.cz` which includes all major libraries in the Czech Republic) and in scientific publishing (in the form of 'discovery portals', often operated by universities for their staff and students (like Masaryk University's `discovery.muni.cz`).

But unlike library catalogues and scientific bibliographies, where the exchange of data and metadata in standardized formats and through standardized APIs is the norm, the lexicographic industry is standardized rather weakly. Data exchange formats exist (such as LMF, the TEI Guidelines Dictionary Chapter and TEI-Lex0, Ontolex/Lemon) but are not widely adopted. Dictionary websites are built to be readable by humans but not by machines. This makes it difficult to build aggregation tools such as dictionary portals. General search engines like Google do not suit this purpose very well because they do not always successfully distinguish between searches for factual content and searches for linguistic information (*tell me about cats* versus *tell me about the word 'cat'*).

4.1.2. The missing infrastructure

Suppose we wanted to build a website which functions as a dictionary portal: it takes a search query from the user and attempts to find relevant matches in many different web-based dictionaries at the same time, either by keeping its own large index of the dictionaries' content (the *metasearch* approach) or by sending concurrent search queries to each individual dictionary website in real time (the *federated search* approach). The portal would be supported by a crawler which indexes third-party dictionary websites. What infrastructure would need to exist on the web for a such a web crawler to be able to usefully index many dictionary websites?

Firstly, we would need to have a metadata standard which would identify a dictionary website as such to a web crawler and which would give the web crawler lexicographically relevant metadata about the dictionary such as:

— Which languages the dictionary contains and what roles they have there: which is the object language (the language the dictionary describes) and which, if different, is the metalanguage (the language in which the descriptions are made); which is the source language (the language of the headwords) and which is the target language (the language of the translations).

— Which subset of the object language the dictionary covers: general vocabulary (LGP[4]), specialized terminology (LSP[5]) and if so, in which discipline or disciplines, phraseology, idioms, a particular dialect and so on.

— Whether the dictionary is intended for encoding or for decoding or for some other lexicographic function.

— What kind of user the dictionary is intended for: native speakers, second-language learners, children, adults and so on.

— What kind of information the dictionary provides: is it mainly an orthographic dictionary, a morphological dictionary, an etymological dictionary?

Secondly, we would need one of two things:

— Either a standard through which the dictionary can tell the crawler which headwords it has entries for and what their URLs are.

— Or a standard through which the portal website can know how to compose a valid search URL and either redirect the user to it or ingest the HTTP response itself.

Thirdly, we would need a standard which allows the crawler to understand the lexicographic semantics of the dictionary entries published on each dictionary website: where in the HTML page each entry begins and ends, where the headwords are, where the definitions are and son.

Most of these desiderata are currently not being met. The necessary standards either do not exist or are not widely adopted. This is why almost no dictionary aggregation tools exist yet, and the few that do exist, such as the European Dictionary Portal and OneLook, are based on large amounts of manual or semi-manual work. The European Dictionary Portal is supported by a community of volunteer curators, each of whom looks after dictionaries in one or more languages. Effectively, the curators do manually what a web crawler should be doing automatically. It is not known how OneLook collects data from the web about the dictionaries it has on its index, but the instructions which it gives to dictionary authors

---

4 Language for General Purposes
5 Language for Specific Purposes

who want their dictionary included in OneLook seem to imply that some amount of dictionary-specific hacking is required for each dictionary.

4.2. Aim of the thesis

In my thesis I will propose a web-based infrastructure for dictionary aggregation which will be an extension and adaptation of several existing web standards:

— Extensions to the web's existing metadata conventions (the `<meta>` tag) and to the *Sitemaps* protocol which will allow online dictionary publishers to announce to the world, in a machine-understandable format, various metadata about their dictionary: what kind of dictionary it is, what language or languages it contains, where (at what URLs) the individual entries can be found and so on.

— An extension to the *OpenSearch* protocol which will make it easy for third-party tools to know how to compose and send search queries to a dictionary website and to receive the search results in a machine-understandable format.

— An extension to the *Schema.org* vocabulary, based on my *inventory of lexicographic information types* from Topic 3, which will allow dictionary publishers to annotate dictionary content on their HTML pages with machine-understandable *HTML Microdata.*

I will argue in my thesis that a loosely coupled infrastructure based on these standards is likely to facilitate openness and aggregation in dictionaries because it will not force dictionary publishers to share more data than they are willing to. Dictionary publishers will only be expected to share machine-readable *metadata* about their dictionaries and their entries, while sharing the actual *data* (or *content*) will be optional. This will follow the same model which has already been tried and tested in library catalogues and in scientific publishing, where what is being shared publicly is metadata about content but not necessarily the content itself.

4.3. Results achieved so far

The inspiration for this topic came from my work on the European Dictionary Portal in the ENeL project. This work brought to light the fact that, in order to be able to build dictionary aggregation tools, we need to put the necessary infrastructure in place first. Since then I have discussed my vision for a web-based dictionary aggregation infrastructure on several occasions (see publications and presentations below) but there no tangible results yet.

4.4. Results to be achieved yet

I will publish my proposals in as an open specification online and also as a peer-reviewed publication. I will implement the proposals on several dictionary websites (including dictionaries published through Lexonomy) as *data providers* and in the European Dictionary Portal as a *data aggregator*.

4.5. Author's publications and presentations

Two presentations in which I summarized my experience from the European Dictionary Portal project and where I outline my vision for a web-based dictionary aggregation infrastructure:

— Michal Měchura (2017) *How (not) to build a European Dictionary Portal.* Final Conference of the European Network of e-Lexicography, Leiden, the Netherlands.
`https://www.youtube.com/watch?v=ORrGelo9ytU`

— Michal Měchura (2017) *Towards a metadata infrastructure for online dictionaries.* Presentation at a meeting of the European Network of e-Lexicography, Budapest, Hungary.
`https://michmech.github.io/pdf/towards-infrastructure.pdf`

Schedule of future work

Autumn 2019 Semester: *Research*

— Research on structural markup in lexicographic XML (Topic 1).

— Research on trees and graphs in lexicography (Topic 2).

— Research on dictionary schemas with a view of compiling an inventory of lexicographic information types (Topic 3).

— Research on web standards for dictionary website aggregation (Topic 4).

Spring 2020 and Autumn 2020 Semesters: *Implementation and publication*

— Publish a specification and implementation of name-value hierarchies (Topic 1).

— Implement graph-augmented trees in Lexonomy for dictionary mapping (Topic 2).

— Implement an inventory of lexicographic information types in Lexonomy (Topic 3).

— Implement aggregation standards on at least one dictionary website and in the European Dictionary Portal (Topic 4).

Spring 2021 Semester

— Finalize the text of my thesis.

References

Aguado-de-Cea, G.; Montiel-Ponsoda, E.; Kernerman, I.; Ordan, N. (2016). *From dictionaries to cross-lingual lexical resources.* In: Kernerman Dictionary News, 24, pp. 25-31.

Atkins, B. T. S.; Kilgarriff, A.; Rundell, M (2010). *Database of ANalysed Texts of English (DANTE): the NEID database project*. In: Proceedings of the Fourteenth EURALEX International Congress, EURALEX 2010.

Atkins, B. T. S.; Rundell, M. (2008). *The Oxford guide to practical lexicography.* Oxford: Oxford University Press.

Benko, V. (2018). *In Praise of Simplicity: Lexicographic Lightweight Markup Language.* In: Proceedings of the XVIII EURALEX International Congress: Lexicography in Global Contexts.

Bogaards, P. (1990). *Où cherche-t-on dans le dictionnaire ?* In: International Journal of Lexicography, 3(2), pp. 79-102.

Fellbaum, C. (1998). *WordNet: an electronic lexical database.* Cambridge: MIT Press.

Joffe, D.; de Schryver, G.-M. (2018) *The TLex Suite User Guide, Version 10.0.10.* `https://tshwanedje.com/docs/TLex%20Suite%20User%20Guide.pdf` (accessed 6 September 2019).

Klimek, B.; Brümmer, M. (2015). *Enhancing lexicography with semantic language databases.* In: Kernerman Dictionary News, 23, pp. 5-10.

Maks, I. (2007). *OMBI: The Practice of Reversing Dictionaries.* In: International Journal of Lexicography, 20(3), pp. 259-274.

Ogbuji, U. (2004). *Considering container elements: When to use elements to wrap structures of other elements.* In: *Principles of XML design,* IBM, `https://www.ibm.com/developerworks/library/x-contain/index.html` (accessed 6 September 2019).

Ó Mianáin, P.; Convery, C. (2014). *From DANTE to Dictionary: The New English-Irish Dictionary.* In: Proceedings of the Sixteenth EURALEX International Congress, EURALEX 2014.

Pemberton, S. (2013). *Invisible XML.* In: Proceedings of Balisage: The Markup Conference 2013. Balisage Series on Markup Technologies, vol. 10.

Polguère, A. (2004). *From Writing Dictionaries to Weaving Lexical Networks.* In: International Journal of Lexicography, 24(7), pp. 396-418.

Wiegand, H. E. (1989). *Der Begriff der Mikrostruktur: Geschichte, Probleme, Perspektiven.* In: Hausmann, F. J; Reichmann, O.; Wiegand, H. E; Zgusta, L. *Wörterbücher: Ein internationales Handbuch zur Lexikographie.* Berlin: de Gruyter, pp. 409-462.

Lexicographic data standards

Lemon (Lexicon Model for Ontologies): `https://lemon-model.net/`

LMF (Lexical Markup Framework): `http://www.lexicalmarkupframework.org/`

TEI (Text Encoding Initiative) Guidelines Dictionary Chapter: `https://www.tei-c.org/release/doc/tei-p5-doc/en/html/DI.html`

TEI-Lex0: `https://dariah-eric.github.io/lexicalresources/pages/TEILex0/TEILex0.html`

Ontolex (Lexicon Model for Ontologies): `https://www.w3.org/2016/05/ontolex/`

Web standards

HTML Microdata: `https://www.w3.org/TR/microdata/`

OpenSearch: `http://www.opensearch.org/`

Schema.org: `https://schema.org/`

Sitemaps: `https://www.sitemaps.org/`

Dictionary writing systems

IDM DPS: `https://www.idmgroup.com/content-management/use-cases.html#Dictionary-Reference`

iLex: `http://www.emp.dk/ilexweb/` (now defunct but available through the Internet Archive's Wayback Machine)

Lexonomy: `https://www.lexonomy.eu/`

TshwaneLex: `http://tshwanedje.com/tshwanelex/`

Dictionary portals

European Dictionary Portal: `http://www.dictionaryportal.eu/`

OneLook: `https://onelook.com/`

Dictionaries

Longman Dictionary of Contemporary English: `https://www.ldoceonline.com/`

New English–Irish Dictionary: `https://www.focloir.ie/`