fLexiCoGraph: Creating and Managing Curated Graph-Based Lexicographical Data

Peter Meyer, Mirjam Eppinger
Institut für Deutsche Sprache, Mannheim
E-mail: meyer@ids-mannheim.de, eppinger@ids-mannheim.de

Abstract

We present the conceptual foundations and basic features of fLexiCoGraph, a generic software package for creating and presenting curated human-oriented lexicographical resources that are roughly modeled according to Měchura's (2016) idea of graph-augmented trees. The system is currently under development and will be made accessible as open source software. As a sample use case we discuss an existing online database of loanwords borrowed from German into other languages which is based on a growing number of language-specific loanword dictionaries (*Lehnwortportal Deutsch*). The paper outlines the conceptual foundations of fLexiCoGraph's hybrid graph/XML data model. To establish a database, XML-based resources may be imported or even input manually. An additional graph database layer is then constructed from these XML source documents in a freely configurable, but automated way; subsequently, the resulting graph can be manipulated and enlarged through a visual user interface in such a way that keeps the relationship to the source document information explicit at all times. We sketch the tooling support for different kinds of graph-level editing processes, including mechanisms for dealing with updated XML source documents and coping with duplicate or inconsistent information, and briefly discuss the browser interface for end users.

Keywords: graph-based dictionaries, editorial process, data modeling, linked data, historical lexicography

1 Introduction

With the rise of Linked Data approaches to lexicography (cf. Bosque-Gil, Gracia & Gómez-Pérez 2016), graph-based data modeling for dictionaries has gained considerable momentum. For the time being, however, there are hardly any ready-made tools that would assist a genuine graph-based lexicographical editing and online publishing process. Much current research has instead concentrated on interlinking existing resources and converting them to RDF triples. Proposals on 'Linked Data-native' models of lexicographical editing have appeared on the horizon only recently (cf. Gracia, Kernerman & Bosque-Gil 2017), and are typically focused on creating standards-compliant graph resources that are well-suited for NLP applications. Older work on graphs in lexicography and lexicology (e.g. Polguère 2012) has largely remained experimental. The software package fLexiCoGraph presented in this paper is intended as a practical tool that enables working lexicographers to create, manually edit, validate and publish human-oriented lexicographical resources, using a graph-based data modeling layer only to the extent necessary and desired, with the help of an easy-to-use administration interface. Later conversion to a Semantic Web-ready format is possible and will be supported by the software.

In order to illustrate our general ideas on data modeling and editing, we will consider, as a running sample use case for the software, an existing online database of German loanwords in other languages (*Lehnwortportal Deutsch*) which is currently in the early stages of being reimplemented with fLexiCoGraph as its new backend. In this use case, graphs are an indispensable tool to represent complex relationships between loanwords and etyma, such as borrowing histories of words (possibly spanning

multiple languages); formation of derivatives and compounds from loanwords in the recipient language; etc. (Meyer 2014; Bowers & Romary 2017). Our choice for the sample use case has mainly expository reasons, however, and is not meant to indicate a preferred area of applicability. We believe that the flexible architecture of the software and data model makes it suitable for a wide range of applications in lexicography; cf. Section 5.

2 Data Modeling

With a clear focus on human-oriented lexicography (instead of computational lexical resources), our approach shares many important features with Měchura's (2016) idea of *graph-augmented trees* in that it does not consist in representing all possible relations between the entities, attributes and so on as graphs, but in using conventional XML documents (henceforth, *resource documents*) as a starting point for the lexicographical process and to superimpose a graph-based data structure (henceforth, *graph component*) only where useful and appropriate for the lexicographical task at hand.

In our sample use case, the resource documents of the *Lehnwortportal Deutsch* represent entries on German loanwords in other languages, taken from different loanword and etymological dictionaries (resources) and encoded in XML. The XML schema typically varies from resource to resource. Resource documents may contain references to still other source files in various digital formats. The graph component[1] is supposed to represent the network of relationships between the words treated in these entries in a cross-resource, unified way. We posit two distinct node types, one for words (more generally, lexical units) and one for word senses, and an array of inter-word relations ('is borrowed from', 'is derived from', 'is a diasystemic variant of', …) as edge types. Given the possible structural and conceptual heterogeneity of the resources, fLexiCoGraph does not impose any restrictions on data modeling. No assumptions regarding the resource documents' XML schemas are made. No particular ontology is presupposed for the graph component, such that the node and edge types needed for a specific application can be configured with respect to their attributes as one wishes.

A central organizational principle of fLexiCoGraph is the separate treatment of two interrelated parts (non-overlapping but interconnected subgraphs) in the graph component, viz. the *source layer* of information as provided by the individual underlying resource documents and the cross-resource *curated layer* representing edited, corrected and annotated lexicographical data that would typically be presented to the end user. In our use case, different loanword dictionaries may provide complementary or even contradictory data on etyma, loanwords, and their relations to each other (source layer); these data must be homogenized and interconnected in manual lexicographical work for the online presentation (curated layer). The hybrid graph/XML data model of fLexiCoGraph thus generalizes Měchura's proposal of "a data structure that allows fragments of entries to be 'shareable', able to appear in multiple entries" (Měchura 2016:98). While the source layer subgraph simply reproduces data already contained in the resource documents, the vertices and edges added on the curated layer enriches the source layer information with arbitrarily complex lexicographical identifications, specifications, abstractions, corrections and generalizations.

The process of graph creation in the source layer is, of course, driven by the resource documents. These files ultimately define conventional "units of presentation" (which might or might not correspond to traditional dictionary entries) that either contain or reference the data to be processed and

1 The implementation currently used for the *Lehnwortportal Deutsch* internally represents relationships between words through a directed acyclic graph modeled in a relational database. This representation is highly limited in scope and generalizability, however, and must be recreated from scratch each time the underlying resource data (set of dictionaries or their entries) changes. See the online documentation and Meyer (2014) for more details.

presented. The mapping process of data and structural information in the XML files onto graph constellations of the source layer is freely configurable for individual resources through either XSLT or a special domain-specific language created for fLexiCoGraph. Each source layer vertex with all of its properties must correspond to an XML element, however large or small, in some resource document; similarly, each source layer edge represents, in a pre-defined way, some kind of structural configuration between the two XML elements corresponding to the vertices connected by the edge. The resulting source layer subgraph *cannot* be edited manually, since it is supposed to be a faithful portrait of information represented in the original resource. In our use case, source layer vertices with their attributes represent XML elements that model words with all their relevant properties as to grammar, part of speech, language/dialect and so on, or that model word senses (and are therefore typically descendant elements of the word elements).

In a similar fashion, a corresponding subgraph on the curated layer is bootstrapped on first import of the resource in a freely configurable and scriptable way. In our sample use case, the curated layer subgraph automatically created from a resource document is often just a replica of the source layer subgraph, where corresponding vertices are interconnected by a dedicated 'source-to-curated' edge type; many automated graph reconfiguration processes take place during bootstrapping, however. Amongst other things, we map homographic German etyma as they appear in different loanword dictionaries (and thus produce multiple etymon graph nodes for homographic words in the source layer) onto only one shared etymon node in the curated layer that is connected to each corresponding source layer node by a 'source-to-curated' edge, thereby formalizing the fact that the different source dictionaries probably refer to the same lexeme. This is a typical way of creating 'links' between different, originally independent resources on the curated layer. In such cases, the nodes or edges in question can be marked automatically by certain searchable attributes (flags) for later lexicographical review; after all, homographic words might still belong to different lexemes. Other flags may mark whether information available at the different source layer nodes linked to one and the same curated layer node is contradictory. The curated layer can be edited manually by the lexicographer and corresponds to the final graph-based lexicographical information the end user will be presented with. In all cases the connection of curated layer data to the original resource data is formally reconstructable in the graph component by following paths from the curated to the source layer.

3 User Interfaces for Data Management and End Users

fLexiCoGraph offers a large number of browser-based administration and editing tools for the working lexicographer. In this paper, only a cursory overview of some of the more important functions and features will be given.

- There are two different, but interrelated presentation and navigation/search modes, both for end users and lexicographers: A template-based one based on resource documents (more or less corresponding to the activity of "browsing through entries"), and one based on the graph component (allowing users to navigate through the graph). Both layers of the latter are navigable, searchable and (for the curated layer) modifiable in an intuitive and interactive visual graph editor. Editing includes deleting and adding edges and vertices, changing their property sets, and merging nodes. Many of the relevant 'data fusion' tasks and problems that arise with graph-editing are well-known from other processes of combining several resources into a homogenized product (cf. Bleiholder & Naumann 2009).
- There are, of course, importing and editing options for resource documents. In the case of changed source data or when individual altered resource documents are imported again, alterations in the source layer of the graph component automatically percolate up to the curated layer, again

triggering flags on nodes and edges where subsequent editorial decisions are needed. This percolation and revision process can be configured by the lexicographer.

- fLexiCoGraph offers a dedicated *graph/XML editor* component that may be used to create and edit *graph-like* XML resource documents. Graph-like XML resource documents mostly or exclusively represent source-layer subgraphs in that they mainly contain (i) elements representing nodes of a graph and (ii) elements explicitly specifying edges between these nodes, using something like an REFID mechanism to refer to the nodes connected by the edge; cf. (Bowers & Romary 2016) for a survey of the state of the art for the sample case of coding etymological relations in XML. Graph-like XML documents can trivially be converted into source-layer subgraphs. In our use case, dictionary entries in loanword and etymological dictionaries must, in certain cases, be excerpted manually into graph-like XML resource documents that specify which words – etyma, loanwords, derivatives of loanwords, etyma of etyma, etc. – in the original dictionary entries (to be modeled as nodes) stand in which relations (borrowing; language-internal variation or diachronic development, etc.; to be modeled as edges) to one another. The original entries may contain information on remarkably complicated borrowing histories spanning multiple languages and leading to graph constellations with long and ramified paths. Manually editing XML documents representing such constellations in a conventional XML editor would be difficult and error-prone. In the graph/XML editor, lexicographers simply construct the 'borrowing graph' for the entry to be excerpted in a visual way by 'painting' nodes and edges between them on a browser canvas. Upon creating a new resource document or selecting an existing one from a list of entries, the GUI displays all related information and allows the creation of new nodes (=words) and edges for the graph of the chosen entry as well as the deletion or edition of existing nodes and edges. Figure 1 shows an example screenshot for our use case. The upper pane is the graph editor

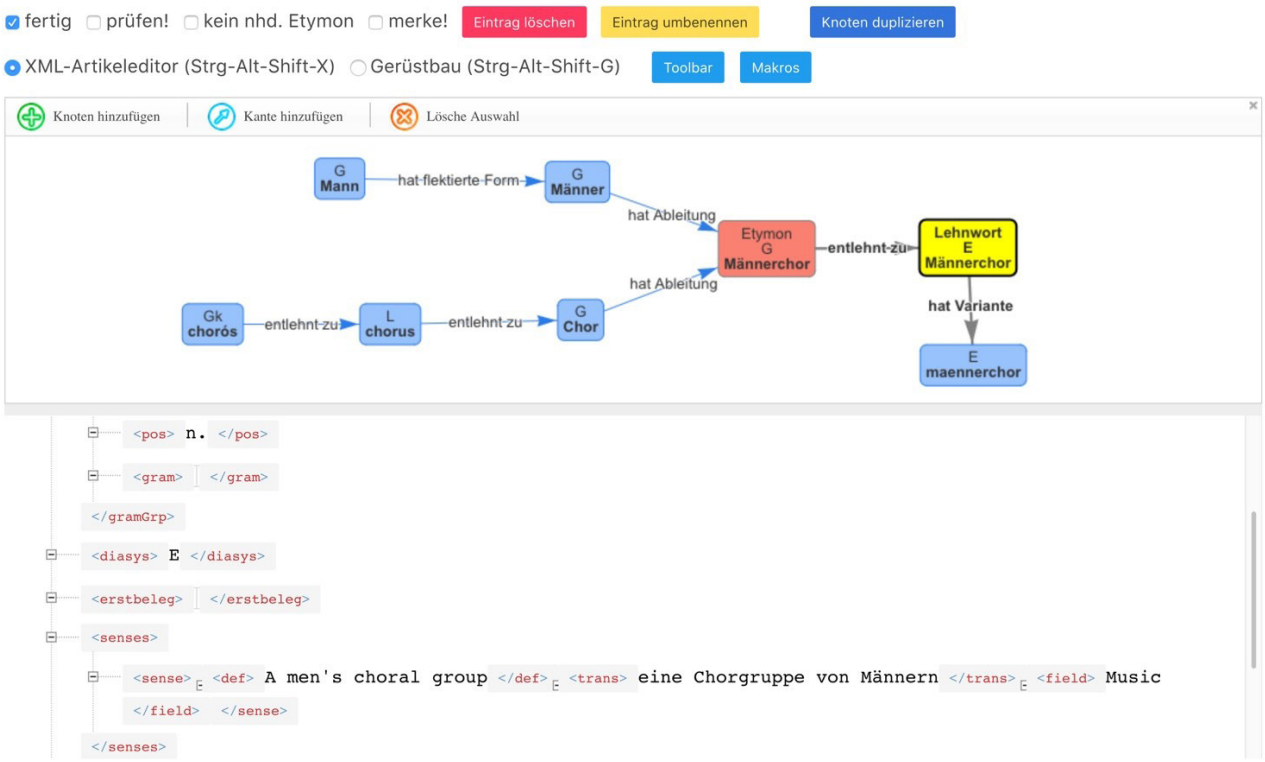

Figure 1: Partial screenshot of a customized graph/XML editor instance; the XML code pertaining to the user-selected yellow node (representing the English loanword *männerchor*) can be edited in the lower pane.

component and shows a manually scalable diagram of the entry-related subgraph. The lower pane, the XML editor component, allows the lexicographer to edit the XML fragment related to the node selected in the upper panel, taking advantage of dropdown lists with pre-defined content or attribute options for certain XML tags (depending on the configuration defined by the user). It is possible to duplicate a node if another shares a lot of XML detail information with it, e.g. in the case of variants that differ only in their word form. There is a separate scaffolding mode that makes fast creation of a first version of a subgraph possible by pasting the original entry into a separate pane (not shown here) and simply selecting words that should be represented as new nodes in the graph.

- Other functionalities include fine-grained user, resource, and editorial rights management, versioning, editorial logging, backup/restore options, and the possibility of assigning user or editorial comments to (parts of) resource documents as well as nodes and edges.
- Complex graph administration tasks can be scripted in the Gremlin graph traversal language.

The browser interface provided by fLexiCoGraph for end users is essentially a stripped-down version of the presentation and search tools at the lexicographer's disposition, without the possibility to alter data and (typically) without direct access to the source layer of the graph component. Note that the online presentation as a whole need not be tied to a network/graph metaphor. Instead, even in the graph-based navigation mode mentioned above the included template-based system allows for defining customized HTML presentations attached to only certain node types (e.g. 'headwords'), and possibly including data from relevant sections of the corresponding resource documents.

4 Software Architecture

fLexiCoGraph is shipped as a cross-platform server application for the Java Virtual Machine with an integrated graph database management system, editing components as well as a web server. In its default configuration, the program runs as a self-contained web application that provides the management interfaces for managing graph-based lexicographical resources and a template-based, freely configurable presentation layer for end users. The software has a pluggable architecture which can also be used

- with a Tinkerpop3-compliant third-party graph OLTP database such as Neo4J;
- with an external XML editor provided it can be customized to exchange XML data with fLexiCoGraph (as is the case at least with most commercial products);
- with a third-party web server provided it implements the Java Servlet specification.[2]

The software is geared towards small to medium size projects. Software development is currently in the prototyping stage; an alpha version with most of the essential features will be available for demonstration in mid-2018. The final product will be made available on the website of the authors' affiliation as open source software – Java source code and binaries for server-side installation – at a later time.

5 Further Areas of Application

We hope that the software package briefly presented in this paper will be useful for a variety of lexicographical tasks, such as those discussed in Měchura (2016) – complex multilingual resources and

2 As an alternative option, fLexiCoGraph may be used only as editing tool; graph data can be exported in a JSON-based representation and published elsewhere.

the treatment of multi-word units – that are best solved with a graph-based approach. A particularly interesting example would be the way resources specifically dedicated to multi-word expressions (MWEs) should be organized. Sets of different entities such as partial hierarchies of ever more specific MWE construction patterns with their slots, fillers and fixed lexical elements as well as distributional contexts exhibit a complex network of interrelations (cf. Steyer and Brunner (2014) for a discussion and Steyer, Brunner and Zimmermann (2013) for a graph-like interactive online presentation). In a graph-augmented model, the vertices representing these entities would be linked to resource documents with full lexicographic descriptions, including corpus examples.

The lexicographical products created with fLexiCoGraph should easily be convertible to a Linked Data format. Tasks such as converting graph data to an RDF format, translating SPARQL queries to database-native Gremlin etc. are indeed conceptually straightforward in fLexiCoGraph. The exact extent to which default tools will be offered in the software presented here remains to be determined.

References

Bleiholder, J. & Naumann, F. (2009). Data fusion. In *ACM Computing Surveys* (CSUR), 41(1), pp. 1-41.

Bowers, J. & Romary, L. (2016). Deep Encoding of Etymological Information in TEI. In *Journal of the Text Encoding Initiative*, 10. Accessed at: http://journals.openedition.org/jtei/1643 [31/03/2018].

Bosque-Gil, J., Gracia, J. & Gómez-Pérez, A. (2016). Linked Data in Lexicography. In *Kernerman Dictionary News* 24, pp. 19-24. Accessed at: http://kdictionaries.com/kdn/kdn24.pdf [31/03/2018].

Gracia, J., Kernerman, I. & Bosque-Gil, J. (2017). Toward Linked Data-Native Dictionaries. In I. Kosem, C. Tiberius, M. Jakubíček, J. Kallas, S. Krek, V. Baisa (eds.) *Electronic lexicography in the 21st century. Proceedings of eLex 2017 conference. Leiden, the Netherlands, 19 – 21 September 2017*. Brno: Lexical Computing CZ s.r.o., pp. 550-559. Accessed at: https://elex.link/elex2017/proceedings-download/ [31/03/2018].

Lehnwortportal Deutsch, ed. by Institut für Deutsche Sprache, Mannheim. Accessed at: lwp.ids-mannheim.de [31/03/2018].

Měchura, M. (2016). Data structures in lexicography: from trees to graphs. In A. Horák, P. Rychlý, A. Rambousek (eds.) *Proceedings of the Tenth Workshop on Recent Advances in Slavonic Natural Languages Processing, RASLAN 2016*, pp. 97-104. Accessed at: https://nlp.fi.muni.cz/raslan/raslan16.pdf [31/03/2018].

Meyer, P. (2014). Graph-Based Representation of Borrowing Chains in a Web Portal for Loanword Dictionaries. In A. Abel, Ch. Vettori, N. Ralli (eds.) *Proceedings of the XVI EURALEX International Congress: The User in Focus. 15-19 July 2014, Bolzano/Bozen*. Bolzano/Bozen: EURAC research, pp. 1135-1144. Accessed at: http://euralex2014.eurac.edu/en/callforpapers/Documents/EURALEX%202014_gesamt.pdf [31/03/2018].

Polguère, A. (2014). From Writing Dictionaries to Weaving Lexical Networks. In *International Journal of Lexicography*, 27(4), pp. 396-418.

Steyer, K., Brunner, A. & Zimmermann, Ch. (2013). *Wortverbindungsfelder Version 3: Grund*. Accessed at: http://wvonline.ids-mannheim.de/wvfelder-v3/ [31/03/2018].

Steyer, K. & Brunner, A. (2014). Contexts, Patterns, Interrelations - New Ways of Presenting Multi-word Expressions. In V. Kordoni, M. Egg, A. Savary, E. Wehrli, S. Evert (eds.) *Proceedings of the 10th Workshop on Multiword Expressions (MWE), Gothenburg, Sweden, 26-27 April 2014*. Association for Computational Linguistics, pp. 82-88. Accessed at: http://www.aclweb.org/anthology/W/W14/W14-0814.pdf [31/03/2018].