LeXmart: A Smart Tool for Lexicographers

Alberto Simões[1,2], Ana Salgado[3], Rute Costa[3],
José João Almeida[2]

[1] 2Ai – Instituto Politécnico do Cávado e do Ave
[2] Algoritmi, Universidade do Minho
[3] NOVA CLUNL, Universidade NOVA de Lisboa
E-mails: asimoes@ipca.pt; anasalgado@campus.fcsh.unl.pt; rute.costa@fcsh.unl.pt;
jj@di.uminho.pt

Abstract

The digital era has brought some challenges to lexicographers, but it has also brought new opportunities as part of the rise of information technology and, more recently, the emergence of digital humanities. This paper provides a description of LeXmart, the framework that supports the digital development of the Portuguese Academy of Sciences Dictionary. LeXmart is a smart tool framework to support lexicographers' work that offers different types of tools, ranging from a structural editor to a set of validation tools.
Given that the dictionary is stored in eXist-DB, LeXmart is developed on top of its ecosystem, using W3C standard languages, and offering default functionalities offered by eXist-DB, namely a RESTful API.

Keywords: e-lexicography; dictionary; lexical databases; lexicographic framework; XML

1. Introduction

The digital era has brought both challenges and new opportunities to lexicographers on the back of Information Technology and the recently developed Digital Humanities. Most e-dictionaries are now embedded into websites, mobile applications, and digital products, besides being also offered as services. Lexicographers have been using a number of computational tools, e.g., word processors, spreadsheets, and in a few cases, databases for their work. Large publishing houses have developed their own in-house systems, but few have made their applications freely or even commercially available. In these new settings, the lexicographic work had to change its course so as to prepare resources and create formats to achieve the main goals of this era: sharing and reusing dynamic data enabling interoperability by using standards and compatible formats.

This paper provides a description of the LeXmart[1] framework to support lexicographers' work, which underlies the digital development of the Portuguese Academy of Sciences Dictionary (DACL), and focuses particularly on its implementation, database support, structural editor, and reporting tools, which have

[1] http://www.lexmart.eu/

proven to be useful for lexicographers to edit the entries and run control checks on them.

As mentioned above, the concept of a dictionary and its production process has undergone major changes on the back of new technologies. Although we can say this also holds true for Portugal, the fact is that these digital resources continue to be designed and implemented according to the same typographic and editorial conventions of the former print editions, "We still consult dictionaries by going to a particular web site. Dictionaries do not come to us" (Tasovac, 2010: 1), without exploring the possibilities of the digital context (Tarp, 2009; Trap-Jensen, 2018).

This paper is structured as follows: Section 2 presents a small introduction to the DACL and its background, and summarizes the process of its conversion from PDF to the structured format of the Text Encoding Initiative (TEI) Dictionaries Chapter. Section 3 presents LeXmart in detail – this section focuses on three main aspects of the framework: tools for lexicographic work, tools supporting website development and information availability, and a brief discussion of the current RESTful API. Finally, in Section 4, we draw some conclusions about the functionalities of the tool, and conclude with further research avenues, both for the specific case of the DACL and of LeXmart.

2. The Portuguese Academy Dictionary

In Portugal, in spite of the successive attempts of the Academy of Sciences (ACL), only in the 21st century (more precisely in 2001) did the ACL publish a complete dictionary (from A to Z), *Dicionário da Língua Portuguesa Contemporânea*, in a two-volume paper version (the first volume from A to F and the second from G to Z). At that time, the authors decided, for a computational approach, to develop a database using Microsoft Access, and a reporting tool to generate a Word file for the dictionary, which was subject to some minor changes both in content and format before printing. Although the database, or even the work file, would be the best source for future developments, the only media that survived these 18 years was the PDF file that originated those same printed versions. In 2015, some preparatory work for an online Portuguese Academy Dictionary was performed through the *Instituto de Lexicologia e Lexicografia da Língua Portuguesa* (ILLLP) and a database was developed by a team working in Natural Language Processing at the University of Minho[2], which now draws on the participation of IPCA[3] and NOVA CLUNL[4].

The DACL is a general language contemporary dictionary with a descriptive nature and a normative concern. It had a synchronous printed edition and it is addressed to a

[2] The team works with Alberto Simões and José João Almeida (Natural Language Processing of the Computer Science Department), and the consultancy of Álvaro Iriarte Sanromán.

[3] Alberto Simões from IPCA is responsible for the technological support of the new digital ACL dictionary.

[4] The participation of NOVA CLUNL is related to the DACL's transition into the TEI Lex-0 format.

vast audience whose mother tongue is Portuguese. A typical entry includes the following elements: headword, pronunciation, followed typically by some linguistic information (e.g., part of speech), the different meanings, usage labelling, synonyms, antonyms, collocations, etymology, and notes. Examples of usage labelling, cross-references, etc., may also be present. In order to guarantee the interoperability and reusability of dictionary content, during the DACL encoding process, the authors have been participating in the TEI Lex-0 discussion[5], a streamlined version of the TEI Guidelines, simplified and enhanced for regular use.

2.1 Reverse engineering: from a PDF to a structured TEI document

The project started with the automatic conversion of a PDF file into a text format, where each string was annotated with its position on the page and the font face and font size used in the original document. A list of pairs containing font faces and sizes was computed and analysed manually. For example, small caps were used to indicate synonyms and antonyms; very large fonts corresponded to the opening letter of each section of the dictionary; a specific font list was used for phonetic transcription. Unfortunately, most of the document uses the same font face and font size, making it impossible to detect automatically what their role in the entry is. Using this information, a superficial and very rough annotation was performed on the PDF transcription.

The next step resulted in the detailed annotation using a set of rewriting rules. These rules, instead of being applied to font information, were applied to the annotated parts of the document and their content. As a case in point, to detect synonyms and antonyms, rewriting rules searched for the asymptotically equal (≃) or the not asymptotically equal (≄) signs. For other finite lists (e.g., grammatical information), a list of the allowed values was prepared manually. For other annotations, positional information (relating to the other already annotated portions of the document) was used.

In order to make this process easier, and as the headwords of the dictionary entries were easy to detect (with a few exceptions that were fixed manually), the full dictionary was divided into thousands of small documents, one for each dictionary entry. This was useful to ensure that the rewriting process was not applied to entries that had already been validated by the TEI schema.

2.2 XML Database

Different approaches were analysed in order to allow lexicographers to edit each dictionary entry cooperatively. The first option considered was the storage of each XML file in a version control system, such as Subversion or GIT. Lexicographers would use

[5] A contribution to the work developed by the DARIAH-ERIC Lexical Resources group: https://www.dariah.eu/activities/working-groups/lexical-resources/.

an IDE (Integrated Development Environment), such as oXygen's XML Editor[6] or Altova XML Spy[7], in order to create, edit, delete and validate entries. Two main issues were behind the decision not to follow this direction: lexicographers had to use the version control system directly (although it would not have been difficult to teach them how to use it, as there are very intuitive clients for these systems, such as GitKraken[8] or Atlassian SourceTree[9], since there was no regular staff, but rather a dynamic team of volunteers, training sessions would have been very hard to schedule); and the difficulty of making the IDE work in a transparent way, without the need for deep XML knowledge. Although not the main issue, the need to index and search the XML files also made us look for other ways of managing XML files.

The second option was to store the documents in a database. For that, and after searching for some options, the eXist-DB[10] database was chosen. Although there are other interesting databases, eXist-DB developers work closely with oXygen XML Editor developers, which makes it easy to connect and use oXygen to edit files stored in eXist-DB. While we do not intend to have all the lexicographers using oXygen, the fact that both the developers and the project coordinators can use it is a valuable asset.

The choice of using eXist-DB paid off, as it is not just an XML aware database, but a feature rich platform to develop XML based applications, allowing the development of websites entirely with W3C standard XML technologies, e.g., XPath, XQuery and XForms. This was the beginning of LeXmart, as small tools started to be developed on top of eXist-DB and, from tool to tool, an interesting and useful framework was developed.

3. LeXmart: a smart framework to support the lexicographers' work

LeXmart is an open-source web platform created to allow lexicographers to easily edit and publish lexical resources. As noted at the end of the previous section, LeXmart started as a set of small independent tools developed on top of eXist-DB. These tools were later compiled in a common interface, resulting in the framework we are presenting here.

This section starts by discussing other tools available to lexicographers to develop their work; it follows with the description of the tools developed on top of the eXist-DB platform, starting with the end-user features (searching), lexicographic support tools

[6] https://www.oxygenxml.com/

[7] https://www.altova.com/

[8] https://www.gitkraken.com/

[9] https://www.sourcetreeapp.com/

[10] http://exist-db.org/

(creating, deleting and editing entries, validating entries, detecting inconsistencies in the whole dictionary), and content management tools; it then provides a brief description of the available API offered by eXist-DB and what will be made public very soon.

3.1 Dictionary editing tools

With the advent of personal computers, publishers started using software applications to help their work on preparing the material for printed dictionaries. While in some situations authors simply used a standard tool (such as a database management system) to help store the information about each dictionary entry, some large companies developed their own dictionary management tools. There is little information regarding these, as such tools were developed in-house to support the publisher's editorial work, and not as commercial tools.

Using the Internet as the backend for a dictionary management system is not new. The DEB (Horák & Rambousek, 2007) was one of the first examples. At that time, Web 2.0 was already a reality, but the DEB was still developed as a typical CGI (Common Gateway Interface) application. Its entries were stored in a Berkeley DB XML database that although XML-aware lacked most of the new XML database functionalities. The interface was also complex and not easy to use. This project evolved (Rambousek & Horák, 2015), implementing SOAP Web Services to interact between a server (DEB) and a set of clients. The server is responsible for the management of the data, using W3C standards, and specifically its dissemination as linked data. DEBWrite is one of the clients, and acts as a front-end application for lexicographers. In order to offer higher customization on the structure of the dictionary entries, DEBWrite provides an online editor for the dictionary micro-structure that parameterizes the dictionary editor. The resulting editor for the dictionary entries is now more versatile, but the interface stills lacks some usability.

LeXmart has been developed since 2016 (Simões et al., 2016a). More recently, Lexonomy (Měchura, 2017) is a good example of what modern dictionary editing software can look like. Lexonomy, offered both as a service and as a software package, also uses Xonomy as the XML editing software, while SQLite is used as the data backend.

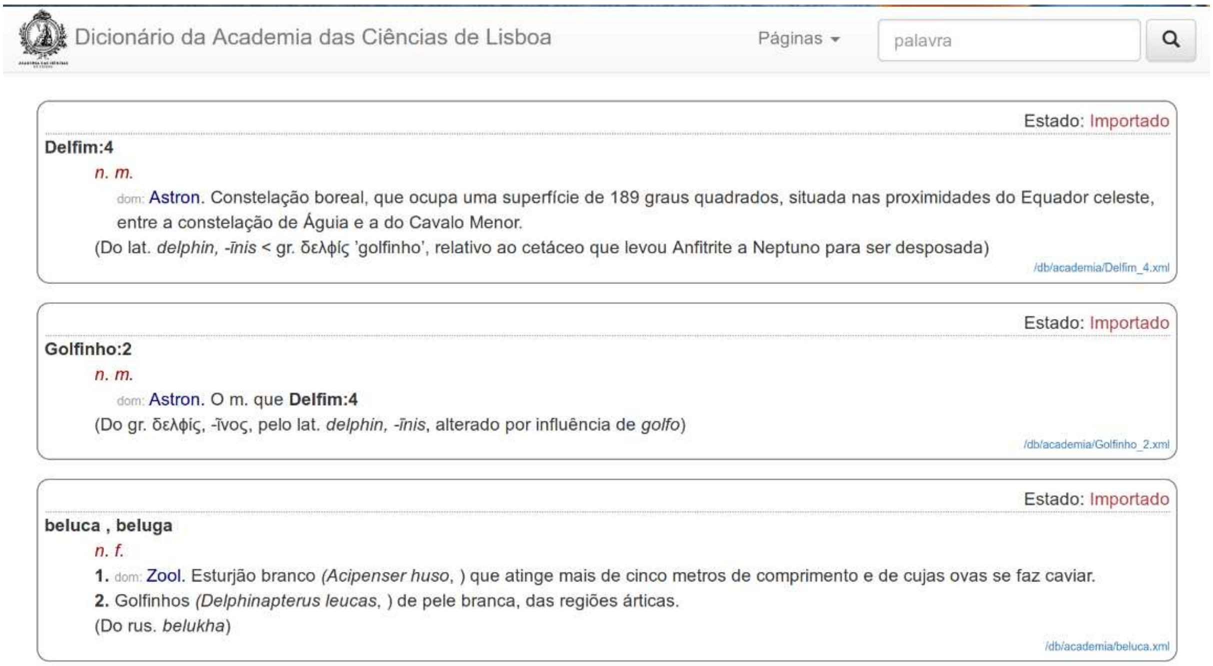
3.2 End-user tools

The DACL is not yet publicly available for end-users. Nevertheless, searching the dictionary is crucial for end-users and lexicographers alike. Therefore, two different approaches were implemented to perform searches: one to search by headword and thus quickly find a definition; and another search by entry content (any part of the entry) enabling a broader search (named *reverse* search), and allowing the user to use the DACL almost as an onomasiological dictionary (Simões et al., 2016b).

The implementation of such queries is quite simple in XQuery, as it allows the search for XML elements containing specific words. Therefore, in the first search type, the query is performed looking up the content of `orth` elements, while in the second search type, the query is performed for the textual content of the whole entry.

The only relevant detail is that eXist-DB uses Lucene as its document database, and therefore the convenient definition of search indexes can make queries much faster.

Presenting the search results is even simpler. With the advent of HTML5, all modern browsers support HTML documents with XML fragments inside (or with HTML with custom tags, if you prefer). Thus, the XQuery script just outputs the entry's XML directly to the browser, which renders it with a custom-defined Cascading Style Sheet (CSS) file. If a user searches, for example, the word *golfinho*, they may obtain all the results where the word *golfinho* occurs, not only in the lemma, but in any section of the lexicographic articles (see Figure 1).



Estado: Importado

Delfim:4

n. m.

dom: Astron. Constelação boreal, que ocupa uma superfície de 189 graus quadrados, situada nas proximidades do Equador celeste, entre a constelação de Águia e a do Cavalo Menor.

(Do lat. *delphin, -īnis* < gr. δελϕίς 'golfinho', relativo ao cetáceo que levou Anfitrite a Neptuno para ser desposada)

/db/academia/Delfim_4.xml

Estado: Importado

Golfinho:2

n. m.

dom: Astron. O m. que **Delfim:4**

(Do gr. δελϕίς, -ῖνος, pelo lat. *delphin, -īnis*, alterado por influência de *golfo*)

/db/academia/Golfinho_2.xml

Estado: Importado

beluca , beluga

n. f.

1. dom: Zool. Esturjão branco *(Acipenser huso,*) que atinge mais de cinco metros de comprimento e de cujas ovas se faz caviar.

2. Golfinhos *(Delphinapterus leucas,*) de pele branca, das regiões árticas.

(Do rus. *belukha*)

/db/academia/beluca.xml

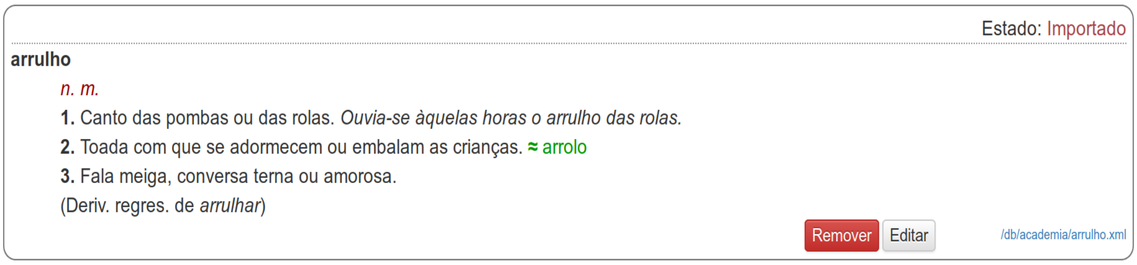
Figure 1: Result of the reverse search for *golfinho* [dolphin] – first three hits.

As can be seen, the entries are shown sorted (proper names are shown first – *Delfim*:4, *Golfinho*:2, and then common names – *beluca*). Meta-information about the entry is also shown (the database URI, e.g. /db/academia/Delfin_4.xml, for the entry document and its revision status, in this case “Importado” [imported]).

3.3 Lexicographic support tools

3.3.1 Entry editor

While using a dedicated XML editor such as oXygen can boost productivity as it contains quite interesting features, it is not user-friendly, and its usage can be rather complex in some situations. In order to allow faster editing, an online editor was developed on top of eXist-DB, based on the Xonomy[11] JavaScript editor. This editor can be accessed by all authenticated users after a headword search. Figure 2 shows the interface presenting the entry for *arrulho* (cooing). Note that there are two buttons, one for editing the entry, and another one for deleting it.



Estado: Importado

arrulho

n. m.

1. Canto das pombas ou das rolas. *Ouvia-se àquelas horas o arrulho das rolas.*

2. Toada com que se adormecem ou embalam as crianças. ≈ arrolo

3. Fala meiga, conversa terna ou amorosa.

(Deriv. regres. de *arrulhar*)

/db/academia/arrulho.xml

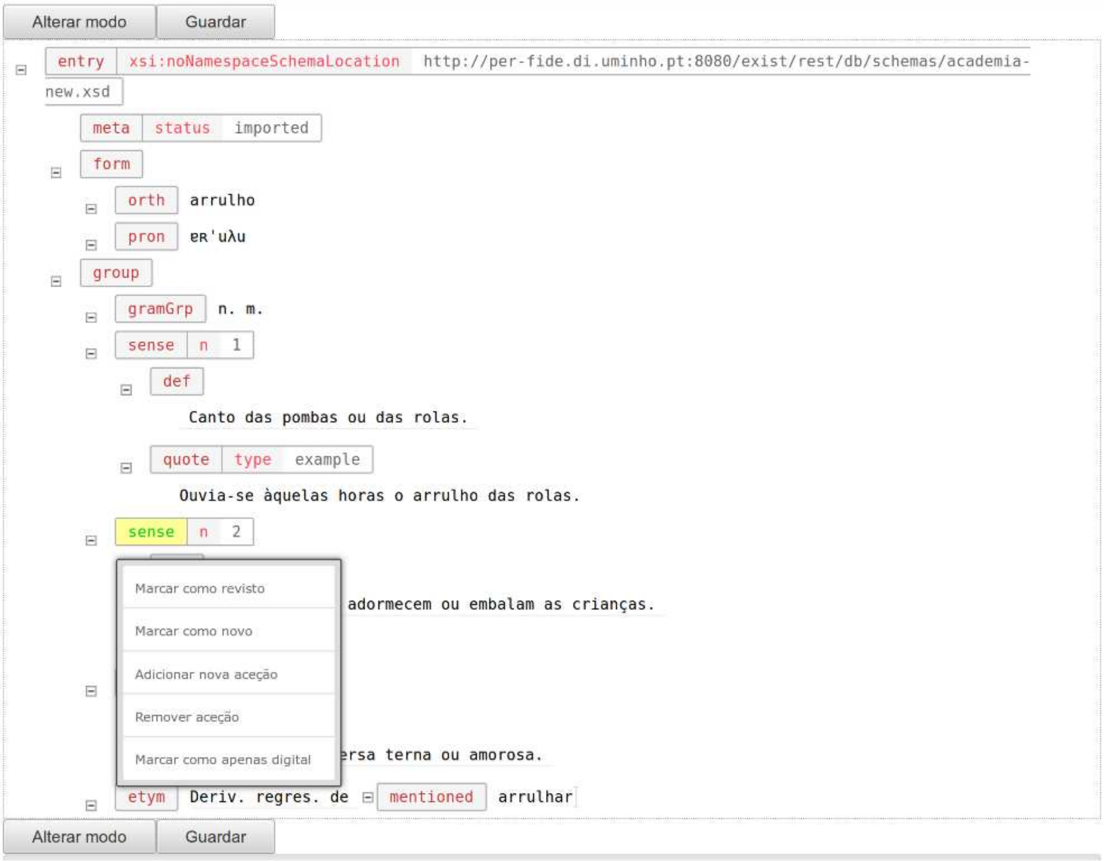
Figure 2: Entry for *arrulho* with authenticated interface for entry editing.

Xonomy is configured by a JavaScript data structure, annotated with some JavaScript functions, that specifies the allowed XML structure, and enables the configuration of drop-down menus to insert, remove, or adapt parts of the entry. The documents edited by Xonomy are fetched and stored using AJAX (Asynchronous JavaScript and XML) calls to the eXist-DB RESTful API.

While Xonomy has its own limitations to support some validation aspects, the XML is internally rewritten to a non-standard XML format, which Xonomy is able to understand and manage correctly. When the lexicographer saves the entry, this non-standard XML format is again converted into valid TEI.

Figure 3 shows Xonomy working. While its appearance is quite similar to an XML document, it is presented without the visual noise of the opening/closing tags. The elements can also be configured with actions. That same figure shows the menu that pops up when the user clicks on a `sense` tag. This menu allows adding some metadata to the entry (revised or as a new meaning), adding a new sense after the current one, removing completely the selected sense, or marking it as digital only. This flexibility of Xonomy that can thus define different actions directly on tags allows the lexicographer to work without the need to know the TEI structure, or the need to directly write XML elements.

[11] Available at https://github.com/michmech/xonomy/.



```
entry  xsi:noNamespaceSchemaLocation  http://per-fide.di.uminho.pt:8080/exist/rest/db/schemas/academia-new.xsd
  meta  status  imported
  form
    orth  arrulho
    pron  ɐʀ'uʎu
  group
    gramGrp  n. m.
    sense  n  1
      def
        Canto das pombas ou das rolas.
      quote  type  example
        Ouvia-se àquelas horas o arrulho das rolas.
    sense  n  2
      adormecem ou embalam as crianças.
      ersa terna ou amorosa.
  etym  Deriv. regres. de  mentioned  arrulhar
```

Figure 3: Xonomy XML editor on top of eXist-DB.

There is also an option to create a new entry in the dictionary. It validates a word that has not been included yet (and if it has, it requires the lexicographer to force its creation with a different entry number), creating the basic XML structure.

3.3.2 Entry creation and deletion

As shown in the previous section, when searching an entry the result list shows a button to remove that entry. When this button is used, a pop-up asks the user to confirm the deletion and removes the entry if requested. Given that the eXist-DB data is being exported to the filesystem as a collection of XML documents that are being stored in a GIT repository (once a day), there is a comprehensive backup of all the changes and deletions, allowing us to recover any mistakenly deleted entry.

Regarding the creation of new entries, there is a small form asking only for the headword. The system automatically searches to see if the word already exists in the dictionary and, if it does, the user is requested to rethink the entry creation, or to explicitly indicate the entry number. If the word is not included in the dictionary, then a new file is created with a boilerplate XML document, with the headword already filled in, and enough structure for the lexicographer to start writing the definition right away.

3.3.3 Meta annotation

Although not directly a developed tool, the annotation of entries or parts of entries with metadata is extremely relevant in order to allow the lexicographers to organize their work.

For example, marking the editing status of an entry is extremely important. For this reason, the possibility of adding this kind of annotation was created. Initially, an entry has the status "imported" (from the original PDF). New entries are created with the "new" tag. Then "revised" is used when the entry has been revised (it is a completed entry) and, finally, "edited", when only a sense or part of the entry has been edited. These statuses can be inserted at the level of the entry or at the level of the different elements that compose the microstructure (usually, senses).

Another important notation is the "digital only" tag, which only appears at the level of the entry or sense, and signals the senses or entries that will only appear in the digital version of the dictionary (and will be excluded from any paper versions).

3.3.4 Filters and statistics

Dictionaries contain information from different sources: different countries or regions of a country, different domains of knowledge, different register types (colloquial, formal, etc.). All this information needs to be codified in the dictionary, and needs to be coherent across the dictionary.

It is easy to find examples of hand-made dictionaries where different abbreviations are used for the same word, different words are used to catalogue different senses in the same domain, and these are only a couple of very simple examples. Using computer tools to assist on the development of a dictionary means these tools should enable some form of consistency check. In part, consistency can be easily guaranteed by using pick-up lists in the editor, but when the work stems from an existing dictionary, other tools need to be developed to find already existing inconsistencies.

In order to allow the lexicographer to control precisely this kind of information, LeXmart has tools to create lists of entries for each type of annotation, and to view graphically the distribution of that information about use.

To provide an example of how these tools are used, consider the work on a specific domain of knowledge, such as biochemistry. While lexicographers are able to construct the entries, and check their structure and completeness, they might not be apt to evaluate the quality of the definitions, or even to write them in the first place. The possibility of filtering the dictionary by a specific area of knowledge allows the lexicographer to export all the entries from that area into a PDF file and send it to an expert in that area. This same type of approach can be used for geographic variants. It is not likely that a Portuguese lexicographer is completely sure about information regarding words imported from Brazil, Angola or Mozambique.

As for statistics (see Figure 4), LeXmart allows the lexicographer to look at the list of possible values for a specific type of markup and understand if there are duplicates (with different forms) or look at a graph and realize whether a specific area of knowledge has insufficient entries to be considered as independent (for example, the printed DACL dictionary has a single entry in the cutlery domain).

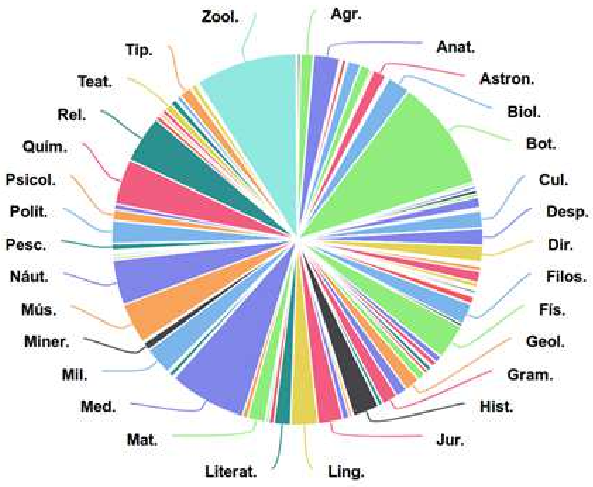


Figure 4: Distribution of areas of knowledge in the DACL.

A final filtering functionality is also available: exporting the entries from a specific text file. Basically, it is possible to upload a text file, where each line includes a headword, and the system will output the entries for those terms.

3.3.5 Reports

In order to understand the evolution process of the lexicographic work, LeXmart allows users to export reports. Currently implemented reports include listing all the new entries that were not present in the printed version of the DACL, the entries from the DACL that were already edited inside LeXmart, and the list of the entries that are marked as finished.

3.3.6 Validation

The eXist-DB database validates the *well-formedness* of the XML syntax, and only allows the storage of valid files. Although it is also possible to configure the database to validate the XML according to a specific schema, that was not the choice as it would limit the storage of files that are being modified, or it would break the full database whenever a minor change was made to the schema. Nevertheless, it is important to know which entries need to be edited and corrected to comply with the defined schema. For that, we created an XQuery validation script. As simple as this script may seem, it took some time to understand the different approaches available for eXist-DB to validate schemas. It takes about 3 minutes to validate the 69K entries outputting an XML document with a report for each failing file. To make reports easier to read, the

XQuery script was tuned to output only the invalid entries. Without it, a full report for all the files would be created.

3.4 Beyond the lexicographic work: content management system

Not directly related to the lexicographic work, a minimal content management system was created in order to allow the creation of ad-hoc pages with relevant information about the dictionary. This system is based on an independent collection where an XML page is created for each page to be published. The pages are edited using TinyMCE[12] 14, a well-known WYSIWYG editor based on JavaScript.

3.5 Portuguese Academy Dictionary RESTful API

Given that the dictionary is stored in eXist-DB, it comes by default with a RESTful API. While the API is currently private, we are working on making the DACL freely available on the web and as soon as that work is finished the API will also be made available. The existence of this API makes bulk editing possible.

In some situations, bulk editing was needed: either some error from the conversion process was detected, or the schema changed to accommodate some new data, or even some changes needed to be made to the entire dictionary. This is still true at the moment as the DACL is progressively being converted from the TEI standard to TEI Lex-0.

For those situations, a practical way to edit each and every document in the database or edit every document that matches a specific pattern is highly relevant. Although the edition can be done entirely in XQuery, having access to a rich language with powerful regular expressions was crucial. With that in mind, a new Perl module was developed (XML::eXistDB::REST) that allows the query of the dictionary, retrieval of documents, and updating their content. This module is under work, but a beta version is already available at the Comprehensive Perl Archive Network (CPAN).

This type of approach has the major disadvantage of not being completely integrated with LeXmart. Nevertheless, its importance makes it worth mentioning.

4. Conclusions and future work

The challenge of converting a paper dictionary into an electronic dictionary is not a new one. This has been done by different teams, and we did it for the *Dicionário Aberto* (*Simões* et al., 2016b) and for the *Dicionário de Sinónimos do Galego* (Gómez Clement et al., 2016). Although we have presented the process of reverse engineering the PDF file and converting it into an electronic dictionary, that is not our main goal. We intend

[12] https://www.tiny.cloud/

to use that dictionary to bootstrap it to live electronically and allow snapshots to paper whenever necessary.

The process of creating a dictionary from scratch or using a previous version as a base can lead to similar problems: how to allow concurrent editing, how to force coherence, how to guarantee regular backups, and other issues. Therefore, we have discussed our approaches to these problems, and how our system was prepared to help lexicographers in their tasks.

Although an interesting set of tools has already been developed, some other requirements made by the lexicographers need to be addressed in the near future:

- Instead of creating HTML reports of each week's work, we intend to create daily and weekly reports of editions, generated as XML documents, imported into another collection. This is a very interesting resource to have, in order to monitor the activity in the dictionary, and to have a log of every change performed.

- Currently, our web application is restricted to authenticated users. In the future, an open interface needs to be available to end-users. Although the simple mechanisms to search for entries are already developed (although restricted), we think there are a couple of other interesting approaches. For example, synonym and antonym annotation can be used to present the dictionary as a graph/WordNet-like structure.

- Formats – either as eBooks or a print version. For that, we expect to create a set of exporting tools, both to ePub format and to PDF. For the latter, we expect to use LaTeX[13] or XSL-FO[14], as these tools enable the automation of the exporting process. This could even allow the dictionary to be exported as different volumes by knowledge area.

- Regarding the framework, LeXmart needs some polishing and should be translated into English. We intend to have the current version available in a GIT repository very soon.

5. Acknowledgements

Research financed by Portuguese National Funding through the FCT – Fundação para a Ciência e Tecnologia as part of the project Centro de Linguística da Universidade NOVA de Lisboa – UID/LIN/03213/2019, and by the European Union's Horizon 2020 research and innovation programme under grant agreement No 731015 (ELEXIS).

[13] https://www.latex-project.org/

[14] Extensible Stylesheet Language Formatting Objects - https://www.w3.org/TR/xslfo20/

6. References

Gómez Clemente, X. M., Guinovart, M. G. & Simões, A. (2016). *Dicionário de Sinónimos do Galego.* Xerais.

Horák, A. & Rambousek, A. (2007). DEB Platform Deployment – Current Applications. In RASLAN 2007: Recent Advances in Slavonic Natural Language Processing, Brno, Czech Re-public. Masaryk University, pp. 3–11.

Měchura, M. B. (2017). Introducing Lexonomy: an open-source dictionary writing and publishing system. In I. Kosem et al. (eds.) *Electronic Lexicography in the 21st Century: Lexicography from Scratch. Proceedings of the eLex 2017 conference, 19-21 September 2017, Leiden, The Netherlands.* Brno: Lexical Computing Ltd., pp. 662-679.

Rambousek, A. & Horák, A. (2015). DEBWrite: Free Customizable Web-based Dictionary Writing System. In I. Kosem, M. Jakubiček, J. Kallas & S. Krek (eds.) *Electronic lexicography in the 21st century: linking lexical data in the digital age.* Ljubljana/Brighton: Trojina, Institute for Applied Slovene Studies/Lexical Computing Ltd., 2015, pp. 443–451. ISBN 978-961-93594-3-3.

Simões, A., Almeida, J. J. & Salgado, A. (2016a). Building a Dictionary using XML Technology. In *5th Symposium on Languages, Applications and Technologies (SLATE'16), vol. 51 of Open Access Series in Informatics (OASIcs).* Germany: Dagstuhl. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, pp. 14:1–14:8.. DOI: http://dx.doi.org/10.4230/OASIcs.SLATE.2016.14.

Simões, A., Iriarte, A. & Almeida, J. J. (2016b). Dicionário-Aberto: construção semiautomática de uma funcionalidade codificadora. In A. Lemaréchal, P. Koch & P. Swiggers (eds.) *Actes du XXVIIe Congrès international de linguistique et de philologie romanes (2013)*, pp. 201–300, Nancy, july. ALTIF. Section 16: Projets en cours; ressources et outils nouveaux.

Tarp, S. (2009). Beyond Lexicography: New Visions and Challenges in the Information Age. In H. Bergenholtz, S. Nielsen & S. Tarp (eds.) *Lexicography at a crossroads. Dictionaries and Encyclopedias Today, Lexicographical Tools Tomorrow.* Bern: Peter Lang AG, International Academic Publishers, pp. 17–32.

Tasovac, T. (2010). Reimagining the Dictionary, or Why Lexicography Needs Digital Humanities. In *Digital Humanities 2010*, pp. 254–256.

Trap-Jensen, L. (2018). Lexicography between NLP and Linguistics: Aspects of Theory and Practice. In *Proceedings of the XVIII EURALEX International Congress: Lexicography in Global Contexts.* Ljubljana: Ljubljana University Press, Faculty of Arts, pp. 25–37.

Websites:

TEI Consortium, eds. TEI P5: Guidelines for Electronic Text Encoding and Interchange. [Version 3.5.0]. [Last updated on 29th January 2019, revision 3c0c64ec4]. TEI Consortium. http://www.tei-c.org/Guidelines/P5/ ([13.07.2019]).

Dictionaries:

DACL: *Dicionário da Língua Portuguesa Contemporânea.* (2001). João Malaca Casteleiro (coord.), 2 vols. Lisboa: Academia das Ciências de Lisboa & Editorial Verbo. New digital edition under revision.