A Graph Visualization Tool for Terminology Discovery and Assessment

Benoît Robichaud

Observatoire de linguistique Sens-Texte (OLST)
Université de Montréal

benoit.robichaud@umontreal.ca

Abstract

This paper presents a Graphical User Interface (GUI) mainly based on a graph visualization device and used for exploring and assessing lexical data found in the DiCoInfo, a specialized e-dictionary of computing and the Internet. Computer visualization devices have been used to present and browse data in many fields, but GUIs for electronic dictionaries have not evolved much. Very few take advantage of the fundamental nature of dictionaries: they are huge and ordered collections of lexical relationships (i.e. *lexical networks*). Graph visualization devices such as intertwined (directed) graphs present themselves as better tools to browse these relationships. They surely are well suited for assessing the consistency of encoded data.

Keywords

Lexical relations, e-dictionary, data visualization, graph model, assessment tool.

1 Introduction[*]

Electronic support to dictionary content management has changed a great deal how data are encoded, managed and retrieved, but little work has been done on innovative ways to give end users ‘a richer experience’. For more than two decades, computer visualization devices have been set up to present and browse data from a multitude of sources and in many fields, but most current electronic dictionaries (*e-dictionaries*) merely continue to replicate the layout of their traditional printed counterparts to display their contents. Aside from image-based dictionaries that are notorious exceptions (for example: the Merriam-Webster’s *Visual Dictionary Online*, QAI’s *The Visual Dictionary*), many advantages of computer capabilities for data visualization have yet to be acquired and adapted in this field.

This paper presents the goals, architecture and usability of a prototypical Graphical User Interface (GUI) primarily based on a graph visualization device and used to browse data and discover knowledge through a subset of selected relations that are found in the DiCoInfo (i.e. *Dictionnaire fondamental de l’informatique et de l’Internet*), an online specialized

[*] We would like to thank M.-C. L’Homme and G. Bernier-Colborne from the OLST for very helpful suggestions and comments on an earlier draft of this paper.

e-dictionary of computing and the Internet. This particular project is part of a larger effort to improve data and knowledge access for language professionals such as technical writers and translators (see L'Homme & Leroyer, 2009; L'Homme et al., 2010).

Its birth is linked to the idea that it was possible to improve the visual and communicative value of dictionary contents using a graph visualization device. First, in displaying the links between the data that appear in field entries: for example, the lexical relationships that exist otherwise among *synonyms*, *derivatives* and *related meanings* of a particular term. Second, in displaying the links between entries that share particular data in some field entry: for example, the relationships among records that mention a particular term as a *derivative* or *related meaning*. Not only do these enhancements seem beneficial, they may be brought together in a single generalized representation that remains neutral with regard to the way the data is accessed. Figure 1 shows the kind of data visualization one can expect to obtain with this approach:

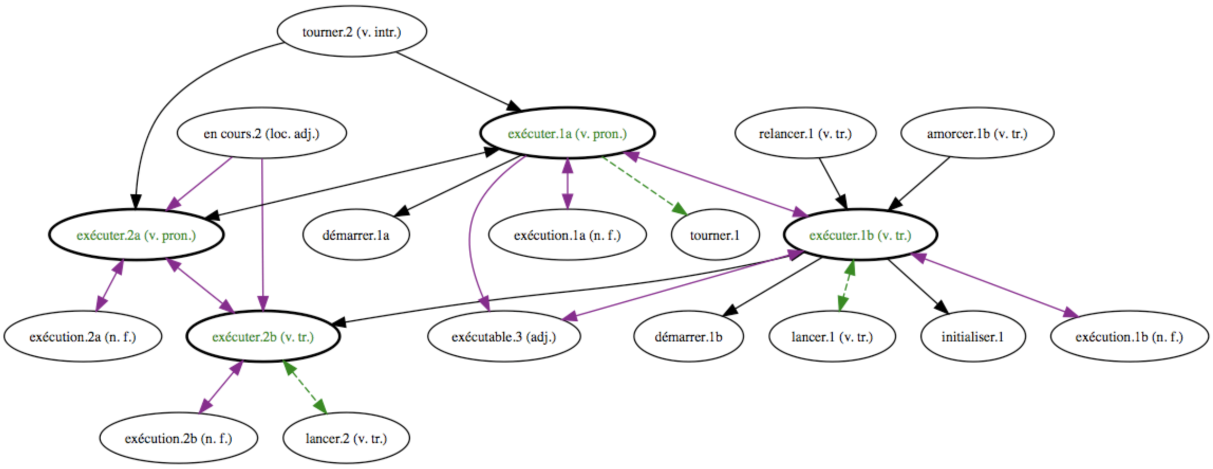


Figure 1: Some of the lexical relations of the polysemous French term 'exécuter'

The actual project was undertaken for two main reasons:

1. We assumed that relationships between terms (perhaps not all, but a large part of them) were likely to be better understood by end users if they were first shown graphically rather than simply listed in tables with textual explanations. In terminology, *taxonomies* and *meronymies* are usually presented in a graphical hierarchy, but other relationships could also lend themselves to a graphical presentation.

2. We also sought to offer a tool for terminologists updating the entries that would help them better assess the consistency of the descriptions. For instance, bidirectional relationships such as *synonyms*, *antonyms*, *derivatives* and *related meanings* could be more easily assessed using a graphical interface.

The rest of the article is organized as follows. Section 2 presents a short overview of traditional GUIs to e-dictionaries and discusses specific drawbacks. It also gives a brief description of a few graph-based GUIs that are found online or downloadable from the Internet. Section 3 first briefly presents the DiCoInfo, and then provides technical details on the architecture and the features implemented so far in our graph-based GUI. Section 4 discusses directions for future work and some of the challenges they raise. Finally, a few concluding remarks are given in Section 5.

2 Old and new ways to explore dictionaries

As previously mentioned, most GUIs to e-dictionaries merely continue to offer traditional outlooks on their contents. Few of them have only the mandatory nomenclature and a display mechanism to view chosen records. Many GUIs offer advanced general and 'by-category' search capabilities that produce (sometimes dynamically) shorter wordlists to help end users access specific contents more efficiently (see for example, *Larousse*, 2011; *Le Petit Robert,* 2011; *OED*, 2011). But wordlists are always presented in the very natural but immutable alphabetically-ordered fashion without showing the links between results. As Manning et al. (2001) mentioned, the basic reason seems to be that, contrary to encyclopedias and thesauri that organize their contents primarily on a conceptual basis, e-dictionaries always compile their contents solely as indexes. Another fundamental reason is simply that they organize and show search results only with respect to field entry organization. A last reason might be that despite the fact that they provide relationships between lexical units, very few encode these relationships formally (however, see Miller, 1993 and Steinlin *et al*., 2005). As Polguère (2009) puts it, the vast majority are simply *text-based* e-dictionaries, that is they only index field entry data and do not organize them otherwise.

Nonetheless, during the last decade, innovative means for exploring e-dictionaries for end users have been proposed. Some of them rely predominantly on lexical networks and offer appealing and interactive graph visualization devices to navigate within their content (e.g., Jansz et al.'s *Kirrkirr*, 2008; The LexiCon Research Group's *EcoLexicon*, 2009; Thinkmap's *Visual Thesaurus*, 2011; logicalOctopus's *Visuwords*, 2011; Vercruysse's *WordVis*, 2011). However, without appropriate additional control options or display features (like drawing options that allow to select relationship types, see Section 3.2), these GUIs can quickly become confusing and users may have trouble untangling all the information presented.

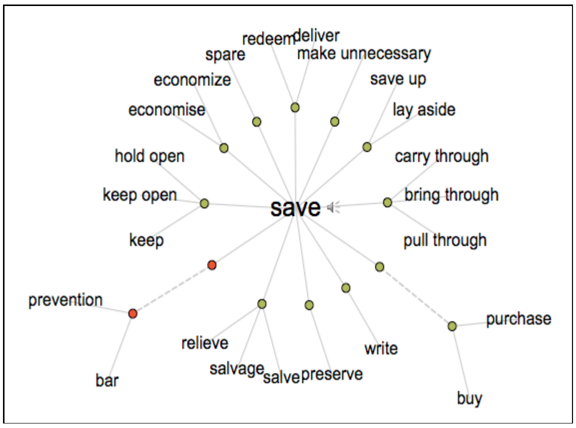


Figure 2
Lexical relations of the English word
'save' from Thinkmap's *Visual Thesaurus*

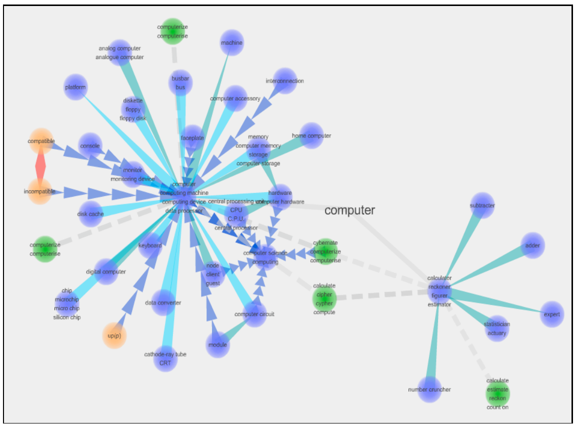


Figure 3
Lexical relations of the English word
'computer' from logicalOctopus's *Visuwords*

3 The DiCoInfo and the DiCoInfo Visuel

As mentioned in the first section, the DiCoInfo is an online e-dictionary that describes terms in the fields of computing and the Internet in French, English and Spanish. It was originally developed as a monolingual tool with the main function of helping end users solve specific knowledge problems associated with this specialized language. From year to year, new languages and functionalities have been added to assist them with tasks such as translation and text production in a second language (see L'Homme et al., 2009).

3.1 The DiCoInfo terminological database

The records of the DiCoInfo are encoded in XML files that are stored in an eXist database management system (see Meier et al., 2011). Apart from the new graph-based GUI presented in Section 3.2 below, end users access and browse the dictionary contents via two main Web interfaces. The first one, called the *static version*, is a compilation of hyperlinked HTML pages that provides the list of all records in the conventional alphabetical fashion. The second one is a *search version* that mimics a search engine and finds the records containing strings (corresponding to parts of words or terms) in specific field entries such as the usual *headword*, *variants* and *synonyms*, but also in other fields that group different sorts (or *families*) of paradigmatic and syntagmatic lexical relationships. These last relationships are formally classified and encoded by means of the *lexical functions* used in the *Explanatory Combinatorial Lexicology* framework (see Mel'čuk et al., 1995 and Mel'čuk, 1996). Both GUIs are implemented using customary XSLT stylesheets that transform the original XML records and put them together in HTML format (see Clark, 1999).

The next subsection describes the architecture of a new graph-based GUI designed for the DiCoInfo. Technical details are provided on the features that have been implemented so far. It is worth mentioning that subsets of lexical functions used in the DiCoInfo were specifically selected for this first version. These encode paradigmatic relationships, namely *hypernyms*, *synonyms*, *antonyms*, *derivatives* and *related meanings*. *Hyponymic* and *meronymic* relationships are not yet incorporated since the data themselves need to be revised and their drawing polished (see Section 4). Lexical functions encoding syntagmatic relationships are also ignored for now as another strategy for displaying them is presently being developed (see Jousse et al., 2011).

3.2 The DiCoInfo *Visuel*

In its current form, the DiCoInfo *Visuel* is a collection of PHP scripts that carry out the following series of tasks: in addition to generating the welcome and result HTML pages, they manage the search options; query the eXist database; receive and analyze the relational data; and last but not least, generate the graph descriptions (to be sent out to a graph drawing device) with a caption and a hyperlinked index of the terms found in the graph. These tasks may be best sketched as a five-step operational cycle that is summarized in Figure 4.

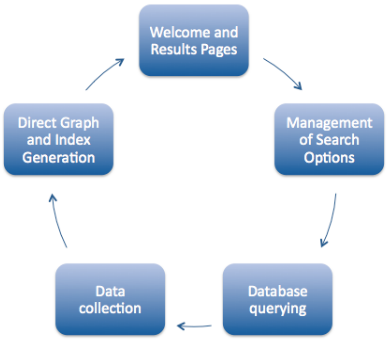
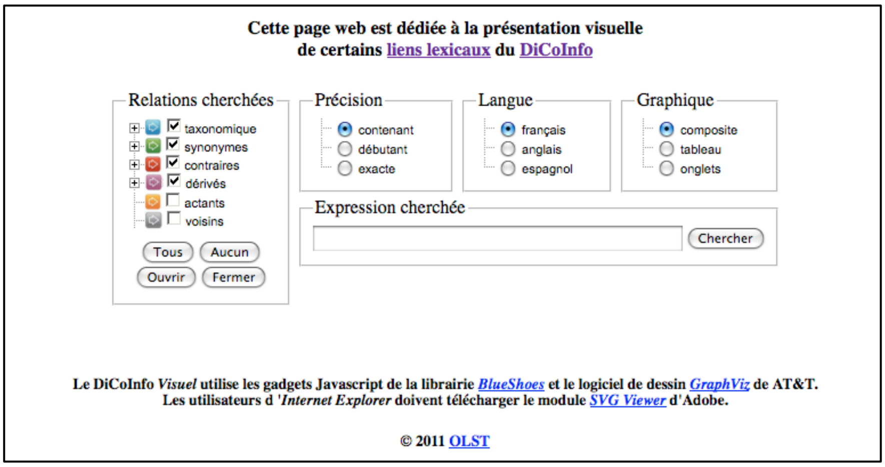

Welcome and Results Pages
Management of Search Options
Database querying
Data collection
Direct Graph and Index Generation

Figure 4: The operational steps in the DiCoInfo *Visuel*

When users access the DiCoInfo *Visuel* without querying it, the main program generates an uncluttered HTML page that welcomes end users (see Figure 5). This page shows usage information, draws the default menu options and finally inserts hyperlinks that point on the one hand to the original HTML static version that contains the word lists of the dictionary; and on the other hand to the sites where the original source code of the Javascript menu framework and the graph drawing tool can be found (respectively, *BlueShoes*, 2011; and

Graphviz, 2011). At this point, users may choose or change some of the search options, type in a query string, and hit the return key for the main program to look up the XML database.



Cette page web est dédiée à la présentation visuelle de certains liens lexicaux du DiCoInfo

Le DiCoInfo *Visuel* utilise les gadgets Javascript de la librairie *BlueShoes* et le logiciel de dessin *GraphViz* de AT&T. Les utilisateurs d '*Internet Explorer* doivent télécharger le module *SVG Viewer* d'Adobe.

© 2011 OLST

Figure 5: Welcome page of the DiCoInfo *Visuel*

The options menu presents four different groups of options. Shown in Figure 6, the first one deals with the different relationships that may be looked for during searches. Note that these relationship options are grouped in families, as are the lexical relationships in the XML database records. The second menu option allows users to define the search precision. It offers to look for data that matches either partially or exactly the string entered. This option allows users to define their queries according to different needs without having to master regular expressions. The third menu option allows searching different parts of the database depending on the language of the search string. The last option is not implemented yet, but will serve to make different renditions of the results (see Section 4).

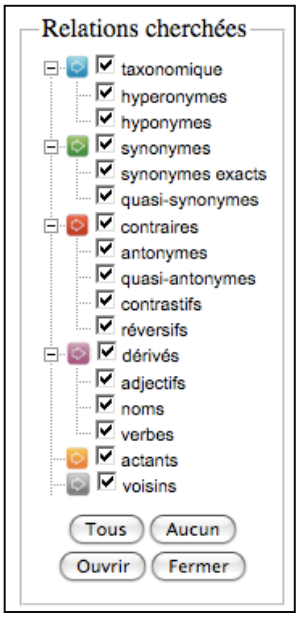


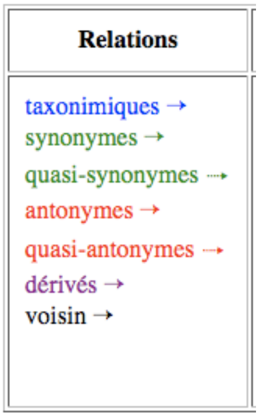
Figure 6: Relationship search options in the DiCoInfo *Visuel*

The next operational step is querying the XML database, but prior to that step the program must put together the queries that are made to the eXist server. One query will be made for each of the (families of) relationships that is selected in the first group of search options. The details of each query are already written in the XQuery language (see Boag et al., 2010) with the exception of the specific values for the search precision, the language option and the searched query string. In other words, for each relationship the program already knows where to look in the dictionary records, and how to format the answer. It is interesting to note that queries for *hypernyms* are recursive and literally walk up and down the relationship paths in the lexical network.

Instantiated XQueries are then submitted to the eXist database server using the XML-RPC protocol (see Scripting News, 2011). For each query, the server returns a collection of very simple XML items of the form: `< link @relation @term1 @term2 />`. Either the searched query string has been found in a *headword*, or in the corresponding relationship field entries. The result items encode the minimal and essential information that the main program needs to know at this point: in the record of '@term1', there exists a relationship of the type

'@relation' that encodes '@term2'. All partial results are then placed in a temporary internal data structure that eliminates the duplicates and records the information to manage the arrowheads of vertices in the future directed graph.

The penultimate operational step is the generation of the graph. The main program first generates its description in the *dot* language (see Ellson, 2011), setting up the display features of the drawing: its general dimensions and orientation; the node list (symbolizing the terms) in which each node has a label, a color, a hyperlink, etc.; and finally the vertices (symbolizing the lexical relationships), each having a color (for its type), a style (for its subtype), its weight and direction, etc. This graph description is then passed to the *Graphviz* drawing software that generates an SVG formatted image (see Dahlström et al., 2011).



Relations

taxonimiques →
synonymes →
quasi-synonymes ⇢
antonymes →
quasi-antonymes ⇢
dérivés →
voisin →

Figure 7: Caption for the relations found

The last operational step of the main program is to generate the HTML results page. This page has the same general display as the welcome page, except that the SVG graph is inserted along with a caption for the relations found (see Figure 7) and an index that may be used to access the traditional search GUI of the DiCoInfo.

The following figures exemplify the kind of graph generated by the DiCoInfo *Visuel*: Figure 8 presents a graph obtained with a recursive query that searches for *hypernyms* of the French term 'disque' (Eng. 'disc'); Figure 9 presents *derivatives* found when searching in French for the substring "exéc" (as in 'exécuter', 'exécutable', etc.); Figure 10 shows a part of the graph involving *synonyms*, *derivatives* and *related meanings* among terms containing the substring "program" in English.

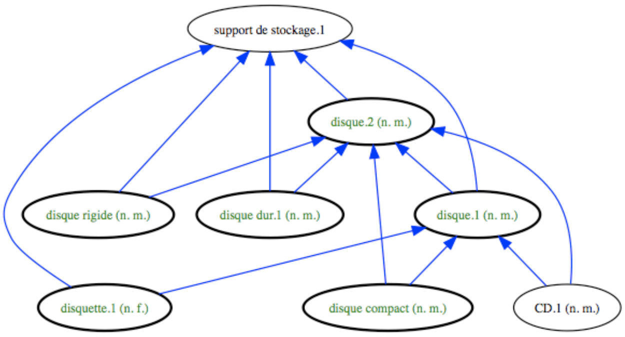


Figure 8: *Hypernyms* of the French terms 'disque'

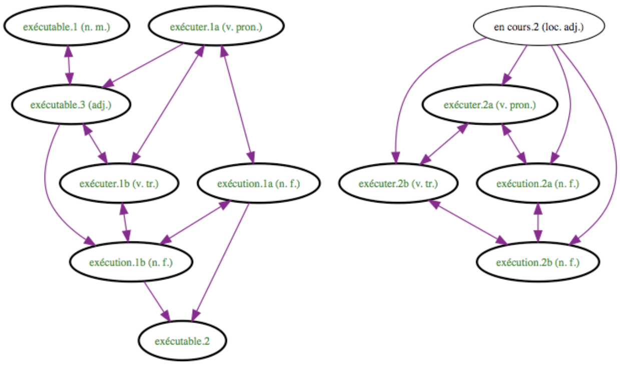


Figure 9: *Derivative* relationships among terms containing the substring "exéc" in French.

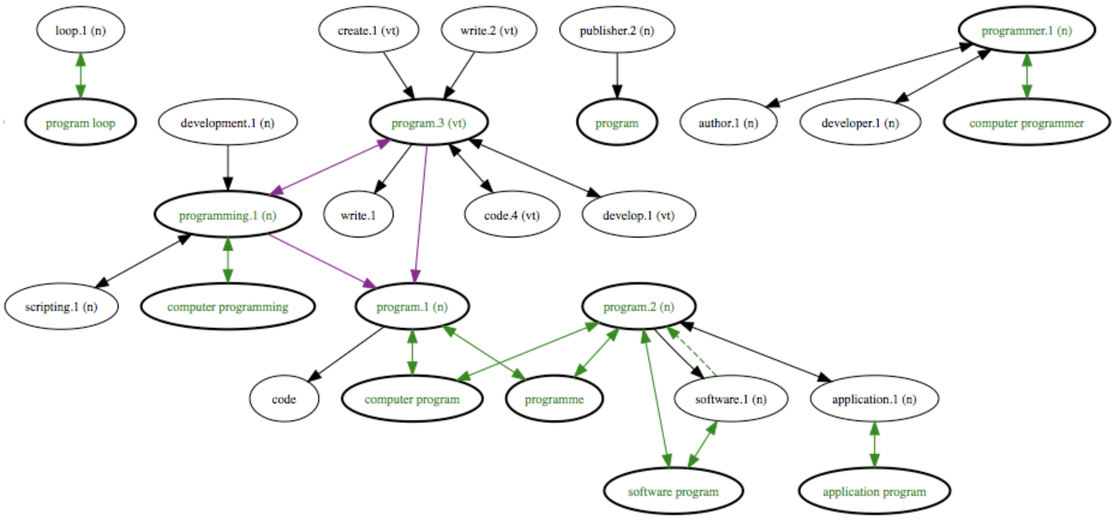


Figure 10: Part of the directed graph with *synonyms*, *derivatives* and *related meanings* of terms containing the substring "program" in English.

4 Improvement and future work

In this section, we discuss three drawbacks noticed during the development and testing of the DiCoInfo *Visuel*. These are dissimilar and will be described independently. For some we propose solutions or improvements. We conclude this section with a brief description of a core feature we intend to implement in the next version.

First, graphs of the DiCoInfo *Visuel* presented so far all have a 'tree' shape, as opposed to the 'spring' shape displayed in other graph-based GUIs mentioned in Section 2. This choice appeared to be a natural one since trees are meaningful and more appropriate for at least subtypes of taxonomic relationships (namely *hypernyms* and *hyponyms*). Other types of relationships seem neutral with respect to this drawing feature. An aesthetical difficulty arises when a particular node has too many direct daughters: these span too widely on the horizontal axis of the tree. One improvement could be made by splitting large sets of daughter nodes into subsets, and distributing them more wisely on the vertical axis with the help of invisible fake nodes. Another solution would be to find the means to mix 'tree' and 'spring' shaped layers in the same graph presentation.

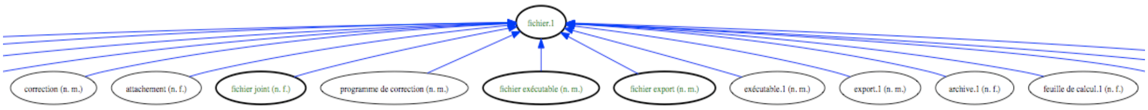


Figure 11: Aesthetical difficulty with nodes having too many direct daughters

Second, as mentioned in Section 2, some queries may simply return too many nodes linked by countless vertices: the entire graph itself becomes extremely difficult to interpret. End users could make a series of more precise queries, but in some cases they may want to visualize all the information anyway. To overcome this 'ergonomic' problem, we plan to implement the last menu options mentioned in Section 3.2 and offer end users the possibility to display the different layers of the resulting graph within table cells or tabs. Another solution would be to display these graph layers in a *Google Gadgets* fashion.

Third, in Section 3.2, we mentioned that setting the search precision option to look for substrings (instead of looking for exact matches) allows to draw richer and more interesting graphs as the XQueries extract large sets of results from the terminological database. Unsurprisingly, this search strategy also finds complex terms by matching their expansion part. For example, a search for 'computer' will locate terms such as 'computer chip', 'computer hacker', 'computer network', and the like. Because shorter terms are preferred as *headwords* and complex ones are mostly encoded as *synonyms*, these terms will appear as orphans if no relationship is found between the *headword* (i.e. 'chip', 'hacker' and 'network') and the base term that corresponds to the expansion (i.e. 'computer').

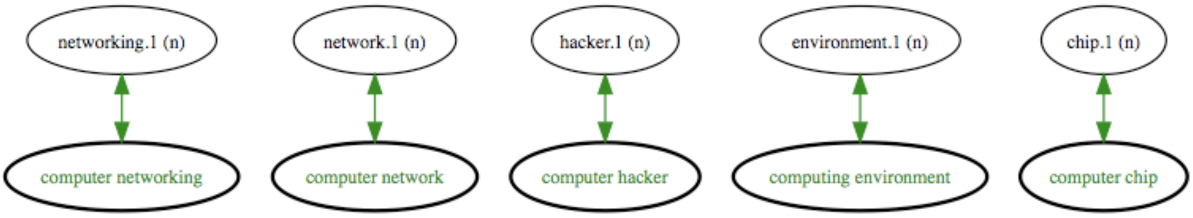


Figure 8: Orphan complex terms found when searching 'computer'

This last problem raises a more significant issue: presently the DiCoInfo *Visuel* is not 'intelligent' and makes no inferences or analogies of any kind. In the next version, in addition to the enhancements discussed above, we intend to build a new architecture of the GUI based on an inference engine as the main program. This new architecture will allow the GUI to draw better graphs, as it will be able to perform the reification of implicit nodes and relationships (see Polguère, 2009). Within this new framework, it will become possible to put in place some inference or analogy mechanisms that will allow generalizing search recursion in the lexical network, and in certain cases compute transitive and deduction closures over the lexical relationships.

5 Conclusion

In this paper, we presented the DiCoInfo *Visuel*, a prototype that puts forward a new method for organizing and visualizing lexical relationships when accessing (specialized) dictionary contents. We addressed the challenge of making dictionary contents accessible and usable for two different types of users (end users and terminologists) through the creation of a single Web GUI. This interface reconciles structured XML data with specialized users' insights when searching, browsing and visualizing terminological information. Our implementation develops a simple and unified solution to the problems of accessing, processing and graphically formatting lexical data in comprehensive ways. Finally, we intend to enhance and expand the software by supplying the actual prototype with an inference engine that we hope will allow to compute lexical analogies and inferences.

Bibliography

Dictionaries

Dictionnaire Le Robert. 2011. *Le Petit Robert de la langue française 2011*.
http://pr.bvdep.com/version-1/pr1.asp [last accessed: 25 May 2011]

LexiCon Research Group. 2011. *EcoLexicon, Terminological Knowledge Base*.
http://ecolexicon.ugr.es/en/ [last accessed: 25 May 2011]

Éditions Larousse. 2011. *Dictionnaires et encyclopédie en ligne*.
http://www.larousse.fr [last accessed: 25 May 2011]

Jansz, K et al. 2008. *Kirrkirr: software for the exploration of indigenous language dictionaries*.
http://nlp.stanford.edu/kirrkirr [last accessed: 25 May 2011]

L'Homme, M.-C., 2011. *DiCoInfo, Dictionnaire fondamental de l'informatique et de l'Internet*.
http://olst.ling.umontreal.ca/cgi-bin/dicoinfo/search.cgi [last accessed: 25 May 2011]

logicalOctopus, 2011. *Visuwords, an online graphical dictionary and thesaurus*.
http://www.visuwords.com [last accessed: 25 May 2011]

Merriam-Webster. 2011. *The Merriam-Webster Visual Dictionary Online*.
http://visual.merriam-webster.com [last accessed: 25 May 2011]

Oxford University Press, 2011. *The Oxford English Dictionary* (OED).
http://www.oed.com [last accessed: 25 May 2011]

QA International, 2011. *The Visual Dictionary*. Éditions Québec Amérique inc.
http://www.ikonet.com/en/visualdictionary [last accessed: 25 May 2011]

Thinkmap, Inc. 2011. *The Visual Thesaurus*.
http://www.visualthesaurus.com [last accessed: 25 May 2011]

Vercruysse, S. 2011. *WordVis, the Visual Dictionary*.
http://wordvis.com/about.html [last accessed: 25 May 2011]

Software

Arn. A. et al. 2011. *BlueShoes: PHP Frameworks & CMS*.
http://www.blueshoes.org [last accessed: 25 May 2011]

Boag, S. et al. 2010. *XQuery 1.0: An XML Query Language* (Second Edition)
http://www.w3.org/TR/xquery [last accessed: 25 May 2011]

Clark, J. (ed) 1999. *XSL Transformations (XSLT)*. W3C Recommendation. Version 1.0
http://www.w3.org/TR/xslt [last accessed: 25 May 2011]

Dahlström, E. et al. 2011. *Scalable Vector Graphics (SVG)*. Version 1.1 (Second Edition)
http://www.w3.org/TR/SVG [last accessed: 25 May 2011]

Ellson, J. et al. 2011. *Graphviz – A Graph Visualization Software*.
http://www.graphviz.org [last accessed: 25 May 2011]

Meier, W. et al. 2011. *eXist : an Open Source Native XML Database*.
http://exist.sourceforge.net [last accessed: 25 May 2011]

Robichaud, B. 2011. *Le DiCoInfo Visuel*. OLST. Université de Montréal.
http://olst.ling.umontreal.ca/dicoinfo/visuel.php [last accessed: 25 May 2011]

Scripting News, Inc. 2011. *The XML-RPC Home Page*.
http://www.xmlrpc.com [last accessed: 25 May 2011]

Other references

Faber, P. et al. 2009. EcoLexicon: A Frame-Based Knowledge Base for the Environment.

Fellbaum, C. 1998. *WordNet: An electronic lexical database*. Cambridge MA: MIT Press.

Collins, C. 2008. *WordNet Explorer: Applying Visualization Principles to Lexical Semantics*. Technical Report. Department of Computer Science, University of Toronto, Canada.

Jousse, A.-L., M.-C. L'Homme, P. Leroyer & B. Robichaud. 2011 (to appear). Presenting collocates in a dictionary of computing and the Internet according to user needs. *Proceedings of the 5th International Conference on the Meaning Text Theory*, Barcelona.

L'Homme, M.-C. et al. 2009. *Le manuel du DiCoInfo*. Département de linguistique et de traduction, Université de Montréal.

L'Homme, M.-C. & P. Leroyer. 2009. Combining the semantics of collocations with situation-driven search paths in specialized dictionaries, *Terminology* 15(2), 258-283.

L'Homme, M.-C., P. Leroyer & B. Robichaud. 2010. Advanced Encoding for Multilingual Access in a Terminological Database – A Matter of Balance, In *Terminology and Knowledge Engineering Conference. Presenting Terminology and Knowledge Engineering Resources Online: Models and Challenges* (TKE 2010), 12-13 August, Dublin.

Manning, C.D., K. Jansz & N. Indurkhya, 2001. Kirrkirr: Software for browsing and visual exploration of a structured Warlpiri dictionary. *Literary and Linguistic Computing*. 16(1): 123-139.

Mel'čuk, I. 1996. Lexical functions: A tool for the description of lexical relations in the lexicon. In Wanner, L. (ed.), *Lexical functions in lexicography and natural language processing*. Amsterdam/Philadelphia: Benjamins. pp. 37–102.

Mel'čuk, I,, A. Clas & A. Polguère. 1995. Introduction à la lexicologie explicative et combinatoire, Louvain-la-Neuve (Belgique): Duculot / Aupelf - UREF.

Mel'čuk, I. et al. 1984-1999. *Dictionnaire explicatif et combinatoire du français contemporain*. Montréal: Presses de l'Université de Montréal.

Miller, B.P., 1993. What to Draw? When to Draw? An Essay on Parallel Program Visualization. *Journal of Parallel and Distributed Computing*, 18(2): 265-269

Miller, G., R. Beckwith, C. Fellbaum, R. Gross & K. Miller, 1993. Introduction to WordNet: An On-line Lexical Database. Felbaum, C. (ed) *WordNet: An electronic lexical database*. Cambridge MA: MIT Press.

Polguère, A. 2009. Lexical systems: graph models of natural language lexicons. *Language Resources and Evaluation*. 43:41-55.

Steinlin, J., S. Kahane, & A. Polguère, 2005. Compiling a ''classical'' explanatory combinatorial lexicographic description into a relational database. *Proceedings of the 2th International Conference on the Meaning Text Theory*, Moscow, pp. 477–485.

Stenmark, D. 1997. *To Search is Great, to Find is Greater: a Study of Visualization Tools for the Web". The result of a class in Human-Computer Interaction*. Internal report.