Tmuse: Lexical Network Exploration

Yannick Chudy[†], Yann Desalle[⋆]
Benoît Gaillard[⋆], Bruno Gaume[⋆], Pierre Magistry[‡], Emmanuel Navarro[†]
[⋆] : CLLE-ERSS, University of Toulouse,
[†] : IRIT, University of Toulouse,
[‡] : INRIA, University of Paris 7

Abstract

We demonstrate an online application to explore lexical networks. Tmuse displays a 3D interactive graph of similar words, whose layout is based on the *proxemy* between vertices of synonymy and translation networks. Semantic themes of words related to a query are outlined, and projected across languages. The application is useful as, for example, a writing assistance. It is available, online, for Mandarin Chinese, English and French, as well as the corresponding language pairs, and can easily be fitted to new resources.

1 Introduction

Although Natural Language Processing applications can not fully replace human abilities to write, read and understand texts, they have proven to be a great assistance for many linguistic tasks. For example, if state of the art Machine Translation (MT) productions can not be considered as accomplished texts, a great variety of Computer Assisted Translation (CAT) software (Trados, OmegaT) help translators work faster, more accurately and consistently. Many writing and reading situations require the extensive use of dictionaries to find or confirm the exact meaning of words to be used in a specific context with specific connotations.

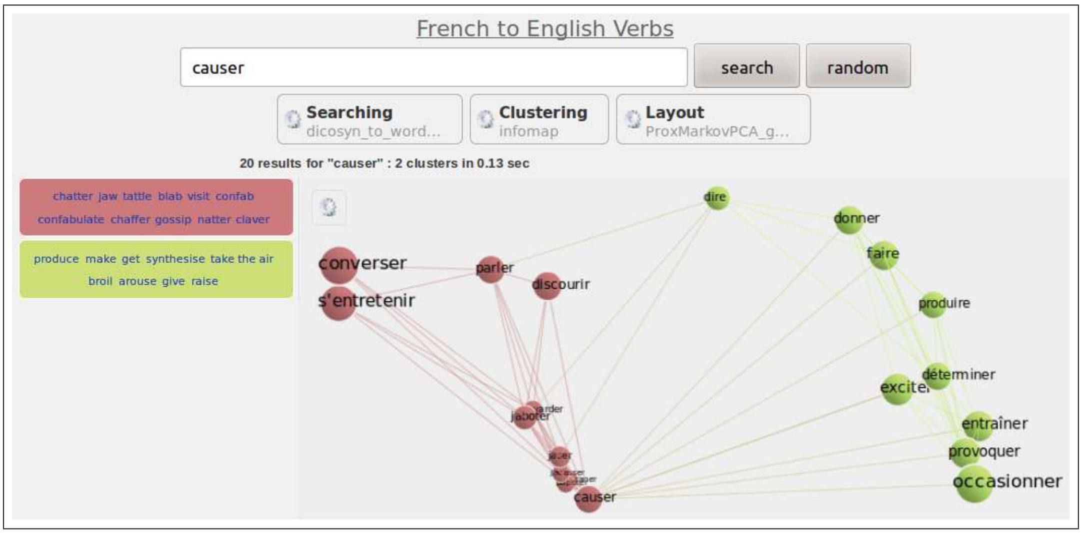
The issue is multiplied when writers manipulate a language for which they are not native. Online dictionaries and thesauri provide the necessary assistance (Linguee, Wordreference, WordNet, Merriam-Webster...), but they can be difficult to make sense of, because, although they provide definitions, subsenses, usage and lists of synonyms, the relations between these informations (semantic similarity of the various synonyms, subsenses) are not directly presented to the user.

Tmuse displays the relations between words that have a meaning similar to that of a query, as shown in Fig. 1. Rather than mere lists, as most dictionaries do, or flat networks of relations [1], Tmuse displays emergent clusters, as semantic fields, and lays out the closest words according to their relative semantic similarity, in a 3D, visually ergonomic presentation. As explained in Navarro et al. (2011), displaying results as a few clusters rather than as long lists is ergonomic, because users find zones of interest at a glance. Beyond the monolingual usage, Tmuse can also help in the cross-lingual case, for instance when the source language is not the user's native language. Tmuse displays the various semantic fields associated with a query word in the source language. Since semantic fields associated with words are not necessarily similar across languages, users might not be familiar with this local semantic structure. Each source semantic field is translated into the target language (user's native language in this example) by a set of target words that are semantically consistent with the source semantic field. This process is called *Proxlation*, as it uses both translation and proxemy in the target language. So, users can grasp, through a set of native language words, the actual meaning of each displayed semantic field, even if it does not constitute a semantic unit of their native language.

The Tmuse exploration tool is based on synonymy graphs and translation bigraphs. The prototype is available online [2] for general English, Mandarin and French, as well as the corresponding language pairs. It can readily be extended to more languages or more specific terminologies, provided the necessary resources.

[1] For example : `homepages.inf.ed.ac.uk/adubey/software/wnbrowser/index.html`

[2] `www.naviprox.net/tmuse`



causer

Searching dicosyn_to_word...

Clustering infomap

Layout ProxMarkovPCA_g...

20 results for "causer" : 2 clusters in 0.13 sec

chatter jaw tattle blab visit confab confabulate chaffer gossip natter claver

produce make get synthesise take the air broil arouse give raise

FIGURE 1 – Example of Tmuse translated Semantic Fields

2 Demonstrated system components

This section overviews the chain of semantic processing components that constitute the backbone of Tmuse, and details resource modelling and theoretical principles underlying each step. Tmuse processes query words to provide a topological description of their semantic landscape. On the basis of a synonymy resource, it first finds a number of *proxemes*, i.e. a set of words that are semantically close to the query. This set of proxemes is then represented as a graph, in 3D. The graph's layout respects the semantic proximity of its vertices, and communities of specifically close words are highlighted. In the bilingual case, proxemy-based sets of translations of these communities are presented in relation with the graph clusters.

2.1 Resource modelling

Synonymy Resources are modelled as graphs $G = (V, E)$, where V is the set of vertices. It corresponds to the resource's lemmas, which are unique. Indeed, the several subsenses of a form are not represented by several vertices, but with the various synonymy connections of the single vertex. A pair of vertices (a, b) defines an edge ($(a, b) \in E$) if and only if a is declared synonymous with b in the resource. The resulting graph is then made reflexive and symmetric.

Wordnet and thesaurus type resources are modelled like the synonymy resources, but edges are drawn within synsets : two lemmas are linked if they belong to at least one common synset. Instead of synsets, the leaves of the thesaurus tree of classes are used.

Translation Resources Translation resources are modelled as bigraphs $B = ((V_1, V_2), E)$, where $V_1 \cup V_2 = V$ is the set of vertices of the graph, V_1 representing the source language lemmas, V_2 the target language lemmas ($V_1 \cap V_2 = \emptyset$). $E \subset V_1 \times V_2$ is the set of edges, which only link lemmas of V_1 to lemmas of V_2 if V_2 is declared translation of V_1 in the resource.

2.2 Word query processing pipeline

2.2.1 Subgraph extraction by random walk

Tmuse uses the Prox algorithm to fetch the N closest words, in the synonymy network, of the query : the *proxemes*.

The Prox algorithm : In a graph $G = (V, E)$, the *Proxemy* of a word to the query is the probability of reaching it by a short random walk of t time steps (Gaume, 2004). Such a random walk can be defined by a Markov chain on V with a $|V| \times |V|$ transition matrix $[G]$ (Bollobas, 2002) :

$$[G] = (g_{u,v})_{u,v \in V},$$

with

$$g_{u,v} = \begin{cases} \dfrac{1}{|N_G^u|} \textbf{if } \{u, v\} \in E, \\ 0 \qquad \textbf{else.} \end{cases}$$

$|N_G^u|$ is the degree of vertex u in G. Let $P_G^t(u \rightsquigarrow v)$ be the probability of a walker starting on vertex u to reach a vertex v after t steps :

$$P_G^t(u \rightsquigarrow v) = ([G]^t)_{u,v}$$

The starting point of a random walk can be generalised to a probability distribution P_0. In that case :

$$P_G^t(P_0 \rightsquigarrow v) = (P_0.[G]^t)_v$$

We call *proxemes* of an initial probability distribution P_0, the vertices of the graph associated with their proxemy. The best proxemes are the ones with the highest proxemy. As shown in Gaume and Mathieu (2007), the "PageRank" approach, biased with a damping factor to the starting point (sometimes called "personalised PageRank"), results in dynamics similar to such short random walks. Its computational cost is however much higher, as it necessitates the knowledge of the whole graph, whereas short random walks only require knowledge of immediate neighbours, at each time-step.

Subgraph Tmuse fetches the N best proxemes of the query. The subgraph induced by this set in the synonymy graph is displayed. In other words the displayed subgraph is made of these proxemes and all the synonymy links they have between themselves.

2.2.2 Graph clustering

State of the art community detection algorithms (Lancichinetti and Fortunato, 2009) are used to partition the extracted subgraph into several semantic zones, materialized on the interface by several colours. We use for instance the Infomap clustering algorithm (Rosvall and Bergstrom, 2008).

2.2.3 Layout

The extracted subgraph, with colour-coded clusters, is displayed in an interactive 3D representation. Vertices are labeled with their lemmas. Their relative positions respect their semantic proximity, thanks to the following algorithm (Gaume, 2008) :

Each vertex u_0 of the subgraph is associated with a proxemy vector P_{u_0} of $|V|$ dimensions : the v coordinate of P_{u_0} is the proxemy between u_0 and the v vertex of the graph : $P_G^t(u_0 \rightsquigarrow v)$.

This models a set of N location in an $|V|$-dimensional space. Two semantically similar words will have similar proxemy vectors and will therefore lie close to each other.

Principal Components Analysis projects this $N \times |V|$- dimensional data set onto $N \times 3$- dimensional data set, that optimally represents its structure.

Clusters computed by the clustering component 2.2.2 are materialised in the layout by different vertex colours. Vertex labels are listed, cluster by cluster, alongside the 3D representation.

2.2.4 Bilingual exploration by *Proxlation*

Like in the monolingual case, the 3D representation describes the semantic topology around the query, with source language words as vertex labels, and the corresponding clusters.

However, the side lists are labelled with the K best translations of the source language clusters, called *proxlations*, and chosen in two steps.

First, Tmuse lists all the translations of all the vertices of the source cluster. Each (target language) translation is weighted according to the number of words of the cluster it translates. This constitutes P_0, a probability distribution vector from which a random walk is launched, on the target language synonymy graph.

The K best proxemes of P_0 are selected as the proxlations of the source language cluster, and appear in the list of the corresponding colour. Selecting proxlations instead of direct translations enables Tmuse to filter out words whose meaning is not consistent with the cluster's semantic theme.

3 System functionalities

3.1 Basic usage

The typical use case of Tmuse is similar to an information retrieval scenario : the user queries a word, and the application replies with relevant lexical semantic information. As described in 2, the application displays a 3D interactive subgraph and lists of related words. Users can make the subgraph turn, zoom on zones of interest, focus on one "semantic field", highlight the actual synonymy links of any word. They can also explore specific meanings by double clicking on words, which launches a new query with this new word. What the interface displays depends on several parameters (number of proxemes, synonymy only, clustering algorithm and layout) that the interested or more advanced user can set.

3.2 Bilingual exploration

In a bilingual mode, users query a word in the source language, and the application displays both the semantic landscape of the query in the source

[3] www.atilf.fr

[4] www.gutenberg.org/ebooks/10681

[5] dict.revised.moe.edu.tw/

[6] cc-cedict.org/wiki/

Name	Language	Type	Reference
Dicosyn	French	synonyms	ATILF & IBM [3]
Wiktionary	French - English	translations	Sajous et al. (2010)
Princetown Wordnet	English	wordnet	Fellbaum (1998)
Roget	English	thesaurus	Gutenberg Projet [4]
Cilin	Mandarin	thesaurus	Mei et al. (1984)
Chinese Wordnet	Mandarin	wordnet	Huang and Hsieh (2010)
MOE dictionary	Mandarin	synonyms	R.O.C Ministry of Education [5]
CEDict	Mandarin - English	translation	dictionary under C.C licence [6]
Authors data	Mandarin - French	translation	own data, to be released soon

TABLE 1 – Resources for Tmuse exploration

language and the proxlations into the target language of each semantic field. Semantic fields are represented by coloured clusters of the extracted source subgraph, their proxlations are displayed in the side lists, with matching colours. Upon clicking on a target word, a new query is launched with the clicked word, on the reverse language pair.

3.3 Resource variations

Users can change the resource of the monolingual application, and also, independently, the source, target and translation resources of the bilingual application. Resources are detailed in Table 1. Results sometimes greatly vary with resource variation. See Gaillard et al. (2011) for an analysis of the similarity of the semantic structure of lexical graphs. Beyond words, Tmuse could be applied to phrases. The computational cost wouldn't be much higher, but one would not only need a phrase translation dictionary, but also phrase synonymy dictionaries. Building such resources could be done by statistical corpus analysis, which would require significant experimental work.

References

Bela Bollobas. 2002. *Modern Graph Theory*. Springer-Verlag New York Inc.

Christiane Fellbaum, editor. 1998. *WordNet : An Electronic Lexical Database*. MIT Press.

Benoit Gaillard, Bruno Gaume, and Emmanuel Navarro. 2011. Invariant and variability of synonymy networks : Self mediated agreement by confluence. In *Proc. of the The 49th ACL-HLT Annual Meeting : 6th TextGraphs workshop*, Portland, Oregon.

Bruno Gaume and Fabien Mathieu. 2007. PageRank Induced Topology for Real-World Networks. *Complex Systems*, to appear :(on line).

Bruno Gaume. 2004. Balades Aléatoires dans les Petits Mondes Lexicaux. *I3 : Information Interaction Intelligence*, 4(2).

Bruno Gaume. 2008. Mapping the form of meaning in small worlds. *Journal of Intelligent Systems*, 23(7) :848–862.

Chu-Ren Huang and Shu-Kai Hsieh. 2010. Infrastructure for cross-lingual knowledge representation - towards multilingualism in linguistic studies. *Taiwan NSC-granted Research Project (NSC 96-2411-H-003-061-MY3)*.

A. Lancichinetti and S. Fortunato. 2009. Community detection algorithms : A comparative analysis. *Phys. Rev. E*, 80(5) :056117.

Jia-Ju Mei, Yi ming Zheng, Yun-Qi Gao, and Hung-Xian Yin. 1984. *TongYiCi CiLin*. Commercial Press, Shanghai.

Emmanuel Navarro, Yannick Chudy, Bruno Gaume, Guillaume Cabanac, and Karen Pinel-Sauvagnat. 2011. Kodex ou comment organiser les résultats d'une recherche d'information par détection de communautés sur un graphe biparti ? In *CORIA'11, Avignon*, pages 25–40. ARIA, mars.

M. Rosvall and C. T. Bergstrom. 2008. Maps of random walks on complex networks reveal community structure. *Proceedings of the National Academy of Sciences*, 105(4) :1118–1123.

Franck Sajous, Emmanuel Navarro, Bruno Gaume, Laurent Prévot, and Yannick Chudy. 2010. Semi-automatic endogenous enrichment of collaboratively constructed lexical resources : Piggybacking onto wiktionary. In Hrafn Loftsson, Eiríkur Rögnvaldsson, and Sigrún Helgadóttir, editors, *Advances in NLP*, volume 6233 of *LNCS*, pages 332–344. Springer Berlin / Heidelberg.

Appendix : Technical details

Tmuse is available online[7]. It runs on a server, hosted in Toulouse, with 4Gb RAM and 3.4 Ghz CPU. The client browser only runs the 3D display. The main memory costs stem from the size and number of the graphs involved. The loaded 27 graphs use 700Mb of memory. As walks lengthen, the number of probabilities to compute and store exponentially increases, so we set a limit to $t = 10$. Clustering algorithms are well-optimised, and applied to only small subgraphs.

[7] www.naviprox.net/tmuse