Enrichissement de la longue traîne d'un réseau lexical grâce à un outil d'évaluation

Mathieu Lafourcade, Alain Joubert

{lafourcade, joubert}@lirmm.fr
LIRMM – Univ. Montpellier 2 - CNRS
Laboratoire d'Informatique, de Robotique et de Microélectronique de Montpellier
161, rue Ada – 34392 Montpellier Cédex 5 – France

Abstract

Lexical networks can be used with benefit for semantic analysis of texts or word sense disambiguation (WSD). Usually strong relations between terms (e.g.: *dog → animal*) are sufficient to help for the task, but quite often, relations of the long tail (e.g.: *dog → necklace, dog → flea*) are necessary. In this paper, our purpose is to acquire such relations by means of online serious games. In the JeuxDeMots project, we decided to relate weights of the relations with the frequency of the propositions of the users. It allows us to acquire first the strong relations, but also those populating the long tail. Furthermore, trying to get an estimation of our network by the users themselves thanks to a tip of the tongue (TOT) software, we realized that they rather tend to favour the relations of the long tail and thus promote their emergence.

Résumé

L'utilité des réseaux lexicaux pour l'analyse sémantique ou la désambiguïsation n'est plus à démontrer. Dans bon nombre d'applications, les relations fortes entre termes (ex : *chien → animal*) sont suffisantes. Il arrive cependant que les relations dites ″de la longue traîne″ (ex : *chien → collier, chien → puce*) soient nécessaires. Le but poursuivi dans cet article est d'acquérir de telles relations par le biais de jeux en ligne. Dans le projet JeuxDeMots, la pondération des relations entre termes est fonction des propositions des utilisateurs : cela nous permet certes d'acquérir en priorité les relations fortes, mais également celles de la longue traîne. De plus, cherchant à faire évaluer notre réseau par les utilisateurs eux-mêmes grâce à un logiciel de type ″Mot sur le Bout de la Langue″ (MBL), nous nous sommes rendus compte que ceux-ci avaient plutôt tendance à privilégier les relations de la longue traîne.

Keywords : Natural Language Processing, serious games, long tailed lexical network, TOT software

Mots-clés : Traitement Automatique du Langage Naturel, réseau lexical, relations de la longue traîne, logiciel de MBL

1. Introduction

Depuis septembre 2007, un réseau lexical de grande taille pour le Français (projet JeuxDeMots JDM, accessible à l'adresse http://jeuxdemots.org) est en cours de construction à l'aide de

méthodes fondées sur des formes de consensus populaire obtenu via des jeux en ligne (Lafourcade et Joubert, 2010). Nous avons ainsi obtenu en un peu plus de 4 ans un réseau lexical de grande taille avec une couverture importante (actuellement 240.000 termes reliés par 1,3 million de relations) représentant une connaissance générale commune. Les relations ainsi créées peuvent être considérées comme des quadruplets : terme origine, terme destination, type et poids de la relation. Le poids d'une relation est fonction du nombre d'utilisateurs qui l'ont proposée. L'interprétation stricte de cette pondération peut être délicate, mais nous pouvons dire qu'elle reflète la ″force″ de la relation. Ainsi, intuitivement, la relation *chien → animal* est plus forte que *chien → collier*, indépendamment de leurs types. Si, en dehors de tout contexte particulier, une relation R1 est évoquée plus rapidement et/ou plus fréquemment qu'une relation R2, alors R1 sera plus ″forte″ que R2.

La plupart des relations présentent dans JDM sont ″directes″ ou ″frontales″ (ex : *chien → animal,* de type ″idée associée″ et dont le poids vaut actuellement 730) ; ce sont celles qui ont été données le plus spontanément par les utilisateurs. Cependant, certaines sont ″indirectes″ ou ″latérales″ (ex : *chien → collier,* de type ″a pour partie″ et de poids actuel égal à 60). Ces dernières relations constituent la *longue traîne*.

Nous cherchons ici à densifier notre réseau, plus particulièrement en augmentant le nombre de relations appartenant à cette longue traîne. Pour cela, nous tirons profit d'un logiciel d'aide à la résolution du problème du ″Mot sur le Bout de la Langue″ (MBL) utilisé en mode tabou, c'est-à-dire en interdisant les relations les plus fortes. Dans une première section, nous rappellerons brièvement les principes de la longue traîne ainsi que sa relation avec la construction de notre réseau. Ensuite, nous introduirons notre logiciel de MBL, dénommé AKI ; nous expliciterons en particulier son utilisation en mode tabou et nous verrons comment cela conduit à la densification de la longue traîne du réseau JDM.

2. Longue traîne d'un réseau lexical

Les réseaux lexicaux, qu'ils soient généraux ou spécialisés, sont maintenant bien connus, en particulier depuis l'émergence de WordNet (Miller *et al.*, 1990). Les relations dans la plupart des réseaux lexicaux ne sont pas pondérées, c'est-à-dire qu'elles sont énumérées, sans aucune indication quant à leur ″force″ ou leur fréquence. L'introduction d'une telle pondération permet de faire la distinction entre les relations fortes et celles qui le sont moins. Les algorithmes de propagation en désambiguïsation pourraient tirer profit d'une telle pondération. Cependant, la question de sa détermination reste délicate : elle est fortement reliée à la fréquence de la relation, mais pas nécessairement uniquement.

Dans les ressources lexico-sémantiques traditionnelles (dont WordNet et ses dérivés) ne se trouvent donc en général que les relations les plus directes. Ceci s'explique essentiellement par les deux modes de construction utilisés pour constituer ses ressources : (a) la construction manuelle, et (b) la construction par extraction statistique depuis un corpus. De plus, la représentation de la longue traîne nécessite un modèle pondéré de relation qu'il est fastidieux de renseigner à la main, et sans doute délicat d'extraire de corpus.

2.1. Principe de la longue traîne

Le concept de longue traîne a été popularisé par (Anderson, 2004) dans le domaine de l'Economie, dans la description de stratégies de ventes de très nombreux articles en petites quantités chacun.

Pour la plupart des termes présents dans JDM, la majeure partie des relations, en nombre, se trouve dans la longue traîne. La figure 1 montre la distribution des poids des relations sortantes du terme *chien*. Une barre verticale a été ajoutée sur cette figure, située vers l'abscisse 61 (approximativement 15% du total de 417 relations) ; elle constitue une indication de la valeur pour laquelle la surface située sous la partie gauche de la courbe (les relations les plus fortes) est égale à celle située sous la partie droite (les relations de la longue traîne). Cela signifie, pour ce terme *chien*, que les 61 relations les plus fortes ont ensemble autant de poids que l'ensemble des 356 relations de la longue traîne. En désambiguïsation, il est communément admis que les relations les plus fortes (celles situées à gauche de la barre) permettent de lever les ambiguïtés dans près de 75 à 80% des cas, les 20 à 25% restants nécessitant les relations de la longue traîne, sous condition bien évidemment qu'elles soient disponibles dans le réseau. Donnons deux exemples de phrases : *Le chien aboie* ne nécessite pas les relations de la longue traîne, alors que *L'axe du chien est bloqué*, invoquant le chien d'une arme, requiert les relations de la longue traîne. La figure 1 présente également une ligne horizontale indiquant le poids moyen des relations. Dans le cas du terme *chien*, cela signifie qu'une centaine de relations ont un poids supérieur à la moyenne, alors qu'environ 300 relations ont un poids inférieur à cette moyenne.

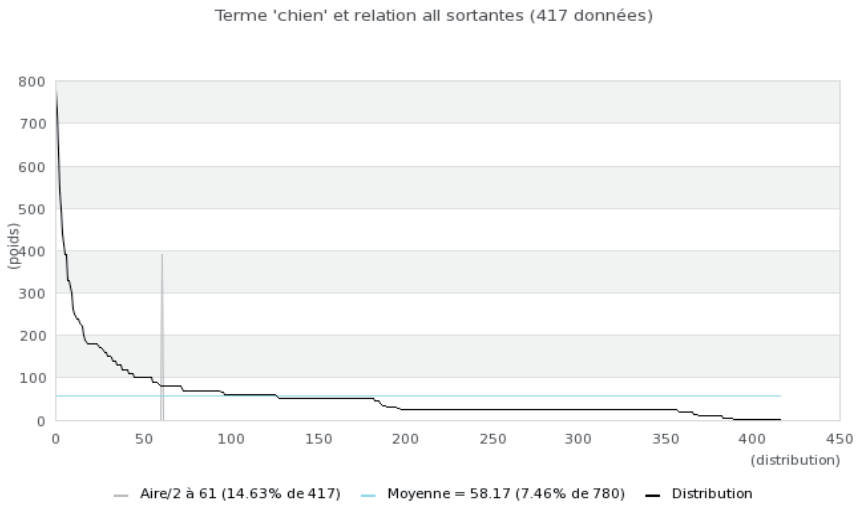

Figure 1 : Distribution des relations sortantes pour le terme chien.

La figure 2 présente la distribution des poids des relations entrantes du terme *chien*. Nous pouvons constater une grande similarité entre ces deux courbes. Toutefois, comme c'est souvent le cas, la distribution des relations entrantes est plus « écrasée » que celle des relations sortantes. De plus, le nombre de relations entrantes est supérieur à celui des relations sortantes, ce qui est souvent le cas pour les termes fréquents ou généraux. Dans les deux cas, l'allure

générale des courbes suit approximativement une loi de puissance, plus probablement une loi de Mandelbrot[1], de la forme $K/(a+bn)^c$.

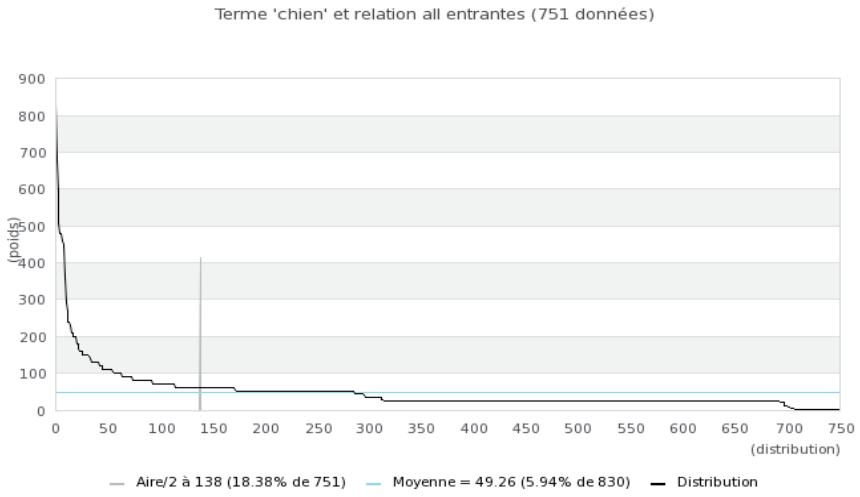

Figure 2 : Distribution des relations entrantes pour le terme chien.

2.2. Construction du réseau lexical

Les principes de base du jeu en ligne JeuxDeMots (JDM), ainsi que la construction incrémentale du réseau lexical à partir d'une base de termes préexistante, ont déjà été publiés, en particulier dans (Lafourcade et Joubert, 2010). Une partie se déroule en asynchrone entre deux joueurs. Pour un même terme cible T et une même consigne (correspondant à un type de relation), nous mémorisons les réponses communes aux deux joueurs. Les validations sont ainsi faites par concordances des propositions entre paires de joueurs.

Ce processus de validation est analogue à celui utilisé par (von Ahn et Dabbish, 2004) pour l'indexation d'images ou plus récemment par (Lieberman *et al.*, 2007) pour la collecte de ″connaissances de bon sens″. A notre connaissance, il n'avait jamais été mis en œuvre dans le domaine des réseaux lexicaux. La structure du réseau lexical ainsi produit repose sur les notions de nœuds et de relations entre nœuds, telle qu'initialement introduite par (Collins et Quillian, 1969) et plus récemment explicitée par (Polguère, 2006). Plus précisément, JDM conduit à la construction d'un réseau lexical reliant des termes par des relations typées et pondérées. Une relation est typée par la consigne donnée aux joueurs, et son poids est fonction du nombre de paires de joueurs qui l'ont proposée.

Un deuxième jeu en ligne, dénommé PtiClic, décrit dans (Joubert *et al.*, 2011), permet de renforcer le réseau produit par JDM. Tout comme JDM, une partie de PtiClic se déroule en asynchrone entre deux joueurs. Un terme cible T, origine de relations, ainsi qu'un nuage de termes provenant de l'ensemble des termes reliés à T dans le réseau produit par JDM, sont proposés à un premier joueur. Plusieurs instructions correspondant à des types de relations sont

1 Dans ce contexte, la loi de Mandelbrot modélise la fréquence d'un terme dans un corpus en fonction de son rang, le terme le plus fréquent étant de rang 1.

également affichées. Le joueur associe, par cliquer-glisser, des termes du nuage aux consignes auxquelles il pense qu'ils correspondent. Ce même terme T, ainsi que le même nuage de mots et les mêmes consignes, sont également proposées à un deuxième joueur. Selon un principe analogue à celui mis en place pour JDM, seules leurs propositions communes aux deux joueurs sont prises en compte, renforçant ainsi les relations du réseau lexical. Contrairement à JDM, les joueurs de PtiClic ne peuvent pas proposer de nouveaux termes, mais sont obligés de choisir parmi ceux affichés. Ce choix de conception permet de réduire le bruit dû aux fautes d'orthographe ou aux confusions de sens.

Ainsi, grâce à l'aide de plus de 2500 joueurs[2], plus de 1,3 million de relations ont été créées, la plupart d'entre elles étant spontanées et donc « frontales ». Les relations « indirectes » sont moins fréquentes, ce qui semble logique compte tenu du mode de création de notre réseau.

3. Le logiciel de MBL : AKI

La question à laquelle nous voulons répondre est la suivante : *pour un terme donné, est-ce que ses relations avec les autres termes du réseau peuvent le caractériser de manière unique ?*

Si la réponse est positive, n'importe quel terme (présent dans le réseau) peut être trouvé via un ensemble réduit de termes indices. Un outil d'aide à la résolution du « Mot sur le Bout de la Langue » (MBL), tel que présenté par (Zock, 2002), peut permettre d'obtenir une estimation de la qualité de notre réseau. Grâce à un tel outil, disponible sur le web, l'évaluation peut être faite de manière permanente avec l'aide d'un grand nombre d'évaluateurs, ceux-ci ne sachant pas qu'ils évaluent.

3.1. Principe

Le logiciel développé, dénommé AKI, outil de MBL, peut également être considéré comme un jeu de devinettes : l'utilisateur essaie de faire deviner un terme au système, en lui fournissant un par un une série de termes indices. Après chaque indice fourni par l'utilisateur, AKI fait une proposition. Elle correspond au terme le plus fortement relié au terme indice dans le réseau. Si AKI a fait la bonne proposition, l'utilisateur la confirme, sinon il introduit un nouvel indice. Le dialogue se poursuit jusqu'à ce que soit AKI trouve le terme cible, soit AKI abandonne et demande à l'utilisateur de lui indiquer la solution.

3.2. Algorithme

L'algorithme sur lequel repose AKI a déjà été décrit dans (Joubert *et al.*, 2011). Nous en rappelons ici les grandes lignes. A partir du premier indice i1, une signature lexicale est calculée : $S(i_1) = S_1 = t_1, t_2, \ldots$ où les t_i sont les termes reliés à i_1 dans le réseau, par ordre d'activation décroissante, c'est-à-dire que t_1 est le terme le plus fortement relié à i_1 dans le réseau. La première proposition faire par AKI est $p_1 = t_1$. Si p_1 est le terme recherché, l'utilisateur valide cette proposition. Si tel n'est pas le cas, l'utilisateur fournit un autre indice. Dans ce cas, on retire i_1 et p_1 de la

2 Le profil moyen des joueurs de JDM est : internaute, entre 30 et 50 ans, de niveau bac + 2 ou au-delà et dans 60% des cas de sexe féminin. Ce profil est toutefois difficile à préciser, et de façon similaire à (Cougnon et François, 2010) pour la collecte d'un corpus de SMS, un biais est certainement introduit par une population de joueurs pas forcément représentative de la population réelle. Cependant, le biais potentiel pour les termes courants n'a pas été observé sur les données.

signature : $S'_1 = S_1 - \{i_1, p_1\}$. En effet, le terme recherché ne peut pas être celui donné en indice, ni la proposition faite par AKI rejetée par l'utilisateur. Après l'introduction du 2ème indice i_2, la signature lexicale S_2 est calculée : $S_2 = (S'_1 \cap S(i_2)) - i_2$ et $S'_2 = S_2 - p_2$. A l'étape n, la signature est $S_n = (S'_{n-1} \cap S(i_n)) - i_n$ et $S'_n = S_n - p_n$ où i_n est le $n^{ème}$ indice donné par l'utilisateur et p_n la $n^{ème}$ proposition faite par AKI.

Avec ce processus, le nombre de termes des signatures successives diminue rapidement. Si la signature se réduit à l'ensemble vide, le système ne peut plus faire de proposition. Dans ce cas, au lieu de faire l'intersection des signatures, nous en faisons l'addition : $S_n = (S'_{n-1} + S(i_n)) - i_n$ et $S'_n = S_n - p_n$. Ce processus de rattrapage conduit rapidement à des signatures volumineuses et, en pratique, il ne peut être réalisé sur plus de deux itérations.

3.3. Réalisation

Le logiciel AKI fait partie du projet JDM. Il est accessible à l'adresse : http://www.jeuxdemots.org/AKI.php.

La figure 3 présente un exemple de partie. Pour le terme cible *chien*, les indices données par l'utilisateur et les propositions faites par AKI sont : *(:isa) animal* → *tigre*, *domestique* → *chat*, *aboiement* → *chien*. Les indices donnés ici sont « frontaux » (ou « directs »).

Figure 3 : Partie d'AKI avec des indices « frontaux » pour le terme chien.

La figure 4 présente un autre exemple de partie. Pour le même terme cible que précédemment, les indices donnés ici par l'utilisateur sont « latéraux » (ou « indirects »). Pour chacun des deux exemples présentés ici, AKI a trouvé le terme cible et donc l'utilisateur approuve en cliquant sur « C'est la bonne réponse ! ».

Figure 4 : Partie d'AKI avec des indices « latéraux » pour le terme chien.

3.4. AKI en mode tabou

De par son principe de construction, correspondant aux associations d'idées spontanées de la part des utilisateurs, notre réseau lexical contient majoritairement des relations « frontales ». Nous souhaitons ici densifier notre réseau en créant ou renforçant les relations « latérales ». Autrement dit, nous voulons pour les termes de notre réseau accroître la population de leur longue traîne.

Pour cela, nous allons faire deviner par AKI un terme cible, en interdisant à l'utilisateur de fournir comme indice les dix termes les plus fortement reliés à ce terme cible dans le réseau, c'est-à-dire en interdisant les indices « frontaux » les plus forts. L'utilisateur est ainsi obligé de fournir des termes indices moins fortement reliés au terme cible et donc appartenant à la longue traîne. Ce processus augmente nécessairement le rappel.

Pour jouer en mode tabou, l'utilisateur doit, sur la page d'accueil d'AKI, sélectionner « Liste de mots » pour découvrir une liste de termes « récemment devinés … ou pas ». L'utilisateur a alors la possibilité de choisir l'un de ces termes pour le faire deviner à AKI, en évitant d'utiliser comme indices les termes tabous qui lui sont affichés (et interdits par le système). Ces termes tabous sont en réalité les dix termes les plus fortement reliés dans le réseau au terme cible choisi. Cela va permettre soit de créer de nouvelles relations appartenant nécessairement à la longue traîne car faiblement pondérées, soit de renforcer des relations existantes mais relativement faibles puisque n'appartenant pas aux dix plus fortes pour ce terme.

La figure 5 montre, pour le terme cible *chien d'arrêt*, les termes tabous que l'utilisateur ne peut pas utiliser pour tenter de faire deviner ce terme cible au système. Remarquons qu'ici il y a moins de dix termes tabous, car le terme *chien d'arrêt* est (encore) relativement peu lexicalisé. L'utilisateur sera donc amené à proposer des indices plus « latéraux », par exemple des noms de race de chien d'arrêt ou des types de gibier chassé par cette méthode de chasse.

Figure 5 : Utilisé en mode tabou, AKI indique les indices que l'utilisateur ne peut pas donner ; ici, ceux pour le terme chien d'arrêt.

Grâce à AKI, les 12.000 parties jouées ont conduit à la création de plus de 50.000 relations de la longue traîne, relations qui n'existaient pas dans notre réseau auparavant. De plus, près de 1200 termes nouveaux ont également été introduits.

3.5. Estimation des performances d'AKI

Les performances d'AKI, en nombre de termes cibles découverts, sont de l'ordre de 75%. La figure 6 montre l'évolution, globalement légèrement croissante, de ces performances.

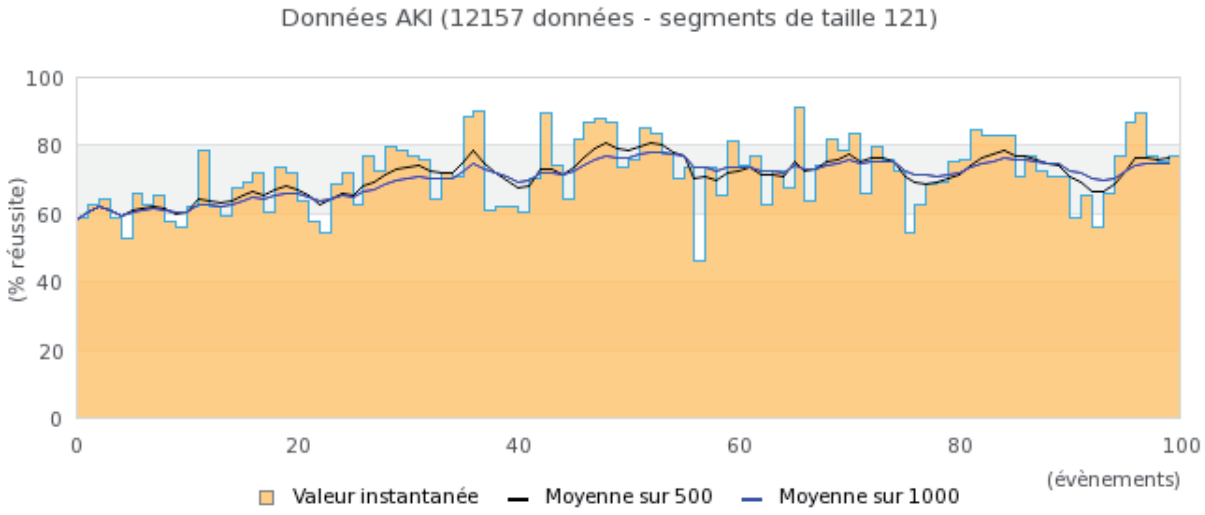

Figure 6 : Evolution du taux de succès d'AKI

A titre de comparaison, nous avons testé auprès de joueurs humains environ 200 parties jouées par AKI. Le taux de réussite des humains a été de seulement 46%, soit moins d'un sur deux, alors que AKI trouve environ trois mots sur quatre !

Une partie trouvée par AKI en mode non tabou, l'est en moyenne au bout de 2,8 indices avec un écart type de 0.6. En mode tabou, ces valeurs sont respectivement de 3,7 et de 1.1.

4. Conclusion

Grâce à l'activité de joueurs, le projet JDM nous a permis d'obtenir réseau lexical de grande taille et avec une bonne couverture lexicale. De plus, ce réseau possède une longue traîne : pour un terme donné, le poids cumulé des 15 à 20% plus fortes relations est analogue à celui des 80 à 85% relations plus faibles. De par le principe de construction de notre réseau, par intersection des propositions entre paires de joueurs, l'émergence de la longue traîne est relativement lente, puisque spontanément les joueurs ont tendance à donner les relations « frontales ». Nous venons de voir dans cet article comment un logiciel de MBL utilisé en mode tabou permet d'augmenter le nombre de relations « latérales » et donc de densifier et accroître la longue traîne, mais également de fournir une évaluation instantanée de sa qualité.

Références

von Ahn L., Dabbish L. (2004). Labelling Images with a Computer Game. *ACM Conference on Human Factors in Computing Systems (CHI)*. pp. 319-326

Anderson C. (2004). The Long Tail, *Wired Magazine*, 12:10, October 2004

Collins A, Quillian M.R. (1969). Retrieval time from semantic memory. *Journal of verbal learning and verbal behaviour*, 8(2), pp. 240-248.

Cougnon L.-A., François T. (2010). Quelques contributions des statistiques à l'analyse sociolinguistique d'un corpus de SMS, *10th International Conference on Statistical Analysis of Textual Data (JADT'10)*, Roma, pp. 619-630

Joubert A., Lafourcade M., Schwab D., Zock M. (2011). Evaluation et consolidation d'un réseau lexical via un outil pour retrouver le mot sur le bout de la langue, *Traitement Automatique des Langues Naturelles (TALN'11)*, Montpellier, pp. 295-306

Lafourcade M., Joubert A. (2010). Computing trees of named word usages from a crowdsourced lexical network. *Investigationes Linguisticae*, vol. XXI, pp. 39-56

Lieberman H., Smith D.A., Teeters A. (2007). Common Consensus: a web-based game for collecting commonsense goals. *International Conference on Intelligent User Interfaces (IUI'07)*. Hawaï, USA.

Miller G.A., Beckwith R., Fellbaum C., Gross D. and Miller K.J. (1990). Introduction to WordNet: an on-line lexical database, *International Journal of Lexicography,* 3 (4), pp. 235-244.

Polguère A. (2006). Structural properties of Lexical Systems : Monolingual and Multilingual Perspectives. *Proceedings of the Workshop on Multilingual Language Resources and Interoperability (Coling/ ACL)*, Sydney, pp. 50-59.

M. Zock. (2002) Sorry, what was your name again, or how to overcome the tip-of-the tongue problem with the help of a computer ? *In SemaNet workshop (Building and Using Semantic Networks)*, Coling, Taipei, pp 107–112.