HyperLex:
Cartographie lexicale pour la recherche d'informations

Jean Véronis[1]
Equipe DELIC, Université de Provence, 29, Av. Robert Schuman, 13621 Aix-en-Provence Cedex 1, France

Abstract

Nous décrivons un algorithme, *HyperLex,* capable de déterminer automatiquement les différents usages d'un mot dans une base textuelle sans utilisation d'un dictionnaire. Cet algorithme est basé sur les propriétés particulières des graphes de cooccurrences, dont nous montrons qu'ils sont de type « petit monde ». La détection des « hubs » et composantes de forte densité de ces graphes permet, contrairement aux méthodes précédemment proposées (vecteurs de mots), d'isoler des usages très peu fréquents (de l'ordre de 1% des occurrences). Nous montrons l'application de cet algorithme à la recherche d'informations sur le Web à partir d'un jeu de mots-test particulièrement ambigus. L'évaluation que nous avons réalisée montre que seuls un très petit nombre d'usages ayant une pertinence thématique ont été omis par l'algorithme. De plus, *HyperLex* permet d'étiqueter automatiquement les usages des mots en contexte avec une précision excellente (97% par rapport à un étiquetage de base [*baseline*] de 73%, pour un rappel de 82%). La sélection des 25 pages les plus pertinentes pour chacun des usages (y compris des usages très peu fréquents) montre elle aussi une remarquable précision (96%). Enfin, *HyperLex* est associé à une technique de représentation graphique permettant à l'utilisateur de naviguer de façon visuelle à travers le lexique et d'explorer les différentes thématiques correspondant aux usages discriminés.

Mots-clés

Désambiguïsation lexicale, recherche d'informations, graphes, petits mondes, interface graphique

1. Introduction

La recherche d'information par mot-clés sur le Web, et dans les grandes bases textuelles en général, se heurte au problème de la multiplicité des usages de la plupart des mots. L'homographie et la polysémie omniprésentes dans les langues introduisent un bruit considérable dans les résultats : ainsi, une recherche sur le mot français *barrage* (anglais : *dam, barrage, barrier, roadblock,* [*police*] *cordon, barricade, blocking,* etc.) verra retourner, au gré des fréquences globales et des heuristiques de classement des moteurs de recherche, des résultats concernant les barrages hydrauliques, les

[1] *E-mail address* : Jean.Veronis@up.univ-mrs.fr

barrages routiers, les matchs de barrage, etc. Extraire les résultats concernant les usages les moins fréquents peut s'avérer particulièrement délicat.

Bien sûr l'utilisateur peut généralement compliquer sa requête en croisant des mots-clés, à l'aide d'opérateurs booléens, mais la requête à construire n'est pas toujours évidente. Ainsi, il ne suffit pas de croiser le mot *barrage* avec le mot *match* pour obtenir les pages concernant les matchs de barrage : de nombreuses pages traitent du thème sans pour autant contenir le mot *match*. Il faudrait alors énumérer les possibilités lexicales et formuler une requête du type *barrage ET* (*jouer OU jeu OU championnat OU rencontre OU football OU basket-ball OU...*), ce qui est peu économique (et peu sûr). De plus, cette technique n'est pas bien maîtrisée du grand public : dans une étude de grande ampleur sur le moteur *Excite,* Spink, Wolfram & Saracevic (2001) montrent que moins de 5% des requêtes utilisent des opérateurs booléens ; de plus, environ 50% d'entre elles sont incorrectes[2]. Moins de 1% des requêtes contiennent des opérateurs imbriqués (comme dans notre exemple précédent). Spink *et al.* concluent même :

> *"For an overwhelming number of Web users, the advanced search features do not exist. The low use of advanced search features raises questions of their usability, functionality, and even desirability, as currently presented in search engines."*

Il semble donc intéressant de revisiter soigneusement l'applicabilité des méthodes de désambiguïsation lexicale aux moteurs de recherche. En effet, une idée reçue semble s'être propagée au cours des dernières années, selon laquelle la désambiguïsation lexicale (et plus généralement les techniques de traitement automatique du langage [TAL]) serait inutile en recherche d'informations (RI), voire en dégraderait les performances. Nous montrerons ci-après que cette affirmation repose sur une interprétation erronée d'articles cités de façon répétitive tels que Voorhees (1999). La présente étude constituera, nous l'espérons, une démonstration de la fausseté de cette position.

Pour être utiles, il faut cependant que les techniques de désambiguïsation lexicale soient suffisamment performantes. De nombreux travaux récents, notamment dans le cadre des compétitions Senseval (Kilgarriff, 1998), ont apporté des améliorations importantes aux techniques et aux ressources disponibles. Toutefois, une des principales difficultés de la désambiguïsation lexicale réside selon nous en amont, dans la liste même des sens qu'utilisent les systèmes. Les dictionnaires classiques sont peu adaptés à la tâche. Ils contiennent la plupart du temps des définitions d'une trop grande généralité (« action de barrer », par exemple), et rien ne garantit qu'elles reflètent le contenu exact du corpus textuel interrogé. Nous avons montré de façon expérimentale la difficulté pour des linguistes de faire correspondre correctement les « sens » d'un dictionnaire et les occurrences d'un corpus (Véronis, 1998). De plus, il resterait à catégoriser automatiquement les documents de la base de textes en fonction des « sens » du dictionnaire, tâche d'une difficulté extrême, qui élude les efforts soutenus de la recherche depuis un demi-siècle (nous renvoyons à Ide & Véronis, 1998, pour un état de l'art détaillé).

Schütze (1998) a proposé une méthode basée sur les « vecteurs de mots » permettant d'extraire automatiquement la liste des « sens » (nous préférons parler d'« usages ») du corpus lui-même, tout en fournissant une technique robuste de catégorisation. Les techniques vectorielles se heurtent toutefois à une difficulté majeure et rédhibitoire : la très grande différence de fréquence entre usages d'un même mot repousse la plupart des distinctions utiles en-dessous du seuil de bruit du modèle.

Nous proposons dans cet article un algorithme radicalement différent, *HyperLex,* capable de déterminer automatiquement les différents usages d'un mot dans une base textuelle sans utilisation d'un dictionnaire. Cet algorithme est basé sur les propriétés particulières des graphes de cooccurrences, dont nous montrons qu'ils sont de type « petit monde » (Watz & Strogatz, 1998 ; Albert & Barabási, 2002). La détection des « hubs » et composantes de forte densité de ces graphes permet, contrairement aux méthodes précédemment proposées (vecteurs de mots), d'isoler des usages très peu fréquents (de l'ordre de 1% des occurrences). Nous montrons l'application de cet algorithme à la recherche d'informations sur le Web à partir d'un jeu de mots-test particulièrement ambigus. L'évaluation que nous avons réalisée montre que seuls un très petit nombre d'usages ayant une

[2] Nos calculs, d'après les tables de Jansen, Spink, & Saracevic (2000).

pertinence thématique ont été omis par l'algorithme. De plus, *HyperLex* permet d'étiqueter automatiquement les usages des mots en contexte avec une précision excellente (97% par rapport à un étiquetage de base [*baseline*] de 73%, pour un rappel de 82%). La sélection des 25 pages les plus pertinentes pour chacun des usages (y compris des usages très peu fréquents) montre elle aussi une remarquable précision (96%). Enfin, *HyperLex* est associé à une technique de représentation graphique permettant à l'utilisateur de naviguer de façon visuelle à travers le lexique et d'explorer les différentes thématiques correspondant aux usages discriminés.

2. Travaux antérieurs

L'application des techniques de désambiguïsation lexicale en RI semble avoir débuté il y a une trentaine d'années, avec les travaux de Weiss (1973), mais ce n'est qu'au début des années 1990 que son application a été expérimentée en vraie grandeur (Krovetz & Croft, 1992 ; Voorhees, 1993 ; Wallis, 1993). Les résultats ont jusqu'ici été assez modestes, et quelques études rapportent même une dégradation des résultats. Ces études, notamment celle, souvent citée de Voorhees (1999), ont sans doute contribué à l'établissement d'une sorte d'idée reçue, comme nous le mentionnons dans l'introduction, selon laquelle la désambiguïsation lexicale et les techniques de TAL en général seraient inutiles et même nocives pour la RI. En fait, cette affirmation est une distorsion des études publiées sur le sujet : si l'on lit soigneusement Voorhees (1999), par exemple, celle-ci insiste largement sur le fait que ce sont des techniques de TAL *imparfaites* qui dégradent les performances dans certaines conditions, et elle est très loin de conclure à l'absence définitive d'intérêt de ces techniques pour la RI. Sanderson (1994) montre de façon expérimentale qu'avec un taux de désambiguïsation correct de 75% (typique de l'état de l'art en matière de désambiguïsation lexicale) les performances en RI se dégradent notablement, les erreurs introduites par le système de désambiguïsation étant pires que l'ambiguïté originale. Sanderson détermine qu'un seuil de 90% de désambiguïsation correcte est nécessaire pour espérer une amélioration des performances en RI. De fait, Schütze & Pedersen (1995), qui utilisent un désambiguïsateur atteignant environ 90% de précision, observent une amélioration de 7 à 14% de leur système de requête.

Il n'est pas anodin que cette dernière étude, qui est l'une des rares à mettre en évidence une influence positive de la désambiguïsation lexicale, ait aussi pour particularité de se passer d'un dictionnaire contenant une liste de sens préétablie : les « sens » sont extraits directement du corpus par une méthode que nous décrirons plus en détail ci-dessous. Selon nous, le dictionnaire est la principale pierre d'achoppement des méthodes de désambiguïsation actuelles : nous avons montré dans une étude à grande échelle dans le cadre de l'action Senseval-1[3] (Véronis, 1998, 2001) que des annotateurs humains avaient les plus grandes difficultés à effectuer le travail de désambiguïsation demandé aux machines. Six informateurs (étudiants en linguistique) devaient étiqueter en parallèle les quelque 3700 occurrences en corpus de soixante mots polysémiques (20 adjectifs, 20 noms, 20 verbes) à l'aide des numéros de sens fournis par un dictionnaire standard (le *Petit Larousse*). L'étude statistique des résultats a montré que l'accord moyen entre paires d'annotateurs était très médiocre : 41% pour les verbes et les adjectifs, 46% pour les noms (une fois soustrait l'effet du hasard). Pour certains mots (par exemple *correct, historique, utile, communication, degré, lancement, station*), le résultat était même virtuellement indiscernable de réponses au hasard. L'analyse détaillée des problèmes montre que, dans la quasi totalité des cas, les entrées du dictionnaire ne contiennent pas suffisamment d'indices de surface pour permettre aux annotateurs de mettre en correspondance les occurrences en corpus avec un sens particulier de façon fiable. Pire, la division même des entrées ne prend que rarement en compte les contraintes distributionnelles des différents sens (nombre et nature des compléments, types de préposition, restrictions de sélection, etc.) — et est en fait très souvent en contradiction avec ces contraintes. Le manque d'ancrage sur les indices et propriétés distributionnelles des mots résulte dans la plupart des dictionnaires en un caractère vague de nombreuses définitions,

[3] Partie pour le français (cf. Segond, 2000).

particulièrement celles de mots abstraits et hautement polysémiques tels que *degré, économie, communication, formation,* etc., qui constituent une part importante de nombreux textes (voir Véronis, 2001, pour une analyse plus détaillée).

Ce résultat expérimental recoupe les remarques constantes effectuées par d'autres chercheurs sur des dictionnaires variés, bien que leurs études aient été généralement plus informelles et moins détaillées (cf. par exemple Ahlswede, 1993, 1995; Ahlswede & Lorand, 1993; Amsler & White, 1979; Bruce & Wiebe, 1998; Jorgensen, 1990). *WordNet,* massivement utilisé en désambiguïsation lexicale, pour des raisons de disponibilité, n'échappe pas à cette règle (cf. Fellbaum, Grabowski, & Landes, 1998) ; s'il offre un réseau riche d'informations lexicales structurées, les divisions de sens qu'il contient sont tout à fait de même nature que celles d'un dictionnaire classique et souffrent des mêmes imperfections.

Comme nous l'avons mentionné, Schütze (1998) s'affranchit du problème en extrayant automatiquement la liste des « sens » du corpus lui-même. Les « sens » sont constitués de groupes (*clusters*) de contextes similaires pour un mot donné, et sont donc totalement définis de manière distributionnelle. Bien que Schütze ne présente pas les choses ainsi, on retrouve donc une idée ancienne. On en trouve la trace chez Meillet (1926) pour qui "le sens d'un mot ne se laisse définir que par une moyenne entre [ses] emplois linguistiques". Wittgenstein (1953) a défendu une position analogue dans les *Philosophische Untersuchungen*, et Harris (1954 : 155-158) l'a adoptée dans son programme linguistique en définissant le sens comme une fonction de la distribution ("meaning as a function of distribution"). Elle est également sous-jacente au travail de Hornby (1942, 1954), qui a eu une influence importante sur la lexicographie britannique. Que les groupes ainsi dégagés constituent réellement des « sens » peut probablement faire débat, et nous utiliserons ci-après, plus prudemment le terme « usage ».

L'implémentation proposée par Schütze s'inspire du modèle d'espace vectoriel bien connu en RI (cf. par exemple Salton & McGill, 1983). Chaque mot est représenté par un vecteur dans cet espace, comme c'est le cas pour les documents et les requêtes en RI ; les dimensions de l'espace sont les différents mots qui peuvent apparaître en contexte avec un mot quelconque du corpus, et la valeur de chaque composante du vecteur correspond au nombre de cooccurrences dans une fenêtre de contexte donnée. Pour reprendre l'exemple du mot *barrage* qui nous a servi d'introduction, et en réduisant les contextes possibles de façon outrancière à deux mots seulement, *eau* et *match,* le vecteur correspondant aurait une représentation dans un espace à deux dimensions du type de celle donnée par la Figure 1.

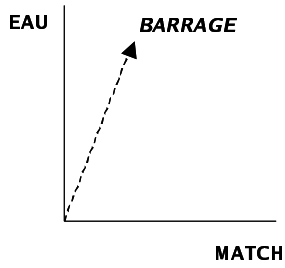

Figure 1. Vecteur du mot *barrage*

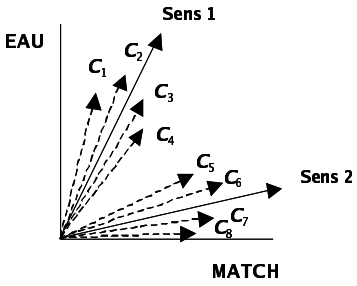

Figure 2. Vecteur-contextes

La représentation vectorielle d'un mot fusionne ses différents usages. Schütze définit donc des vecteurs de contextes, donc chacun est la somme des vecteurs de mots qui apparaissent dans une certaine fenêtre pour chacun des contextes donné (Figure 2). Un algorithme d'agglomération (*clustering*) permet de dégager des groupes de contextes cohérents, qui sont identifiés avec les différents sens du mot concerné. Dans la pratique, les espaces vectoriels ont évidemment des milliers de dimensions, et Schütze utilise une décomposition en valeurs singulières permettant de réduire fortement la dimensionnalité de l'espace (à environ une centaine de dimensions seulement), avant d'appliquer l'algorithme d'agglomération.

L'expérimentation conduite par Schütze sur un ensemble de mots-test montre de très bons résultats, et, comme il a été dit, la technique permet d'améliorer de façon significative les performances d'un système de RI. Toutefois, à part la consommation importante de ressources computationnelle qu'elle implique, elle souffre d'un inconvénient majeur : nos tentatives de réplication montrent que l'algorithme d'agglomération ne peut mettre en évidence que des usages qui sont peu nombreux, relativement équiprobables et fortement individualisés. Schütze utilise d'ailleurs dans son expérience des mots qui satisfont à ces critères, c'est-à-dire des homographes ou quasi homographes équilibrés : *plant, train, vessel,* etc.

La plupart des mots, dont les mots-test que nous utiliserons dans la présente étude, à commencer par *barrage,* font apparaître un ou deux usages prépondérants, suivis d'un nombre plus ou moins grand d'usages de faible fréquence, selon une loi approximative *rang* × *fréquence* ~ *constante* (ceci est déjà apparent dans les comptages de Thorndike & Lorge, 1938 ; voir aussi Krovetz & Croft, 1992). La Figure 3 montre par exemple les fréquences des différents usages du mot *barrage* observées dans un corpus de cinq millions de mots (d'après les données de Reymond, 2002).

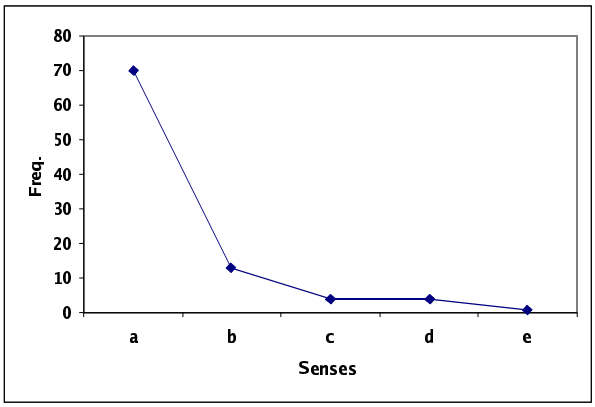

Figure 3. Fréquences des sens de *barrage* relevés en corpus

Les usages de faible fréquence ne sont pas pour autant *rares* pour un locuteur moyen (*match de barrage,* etc.), et sont tout à fait susceptibles de faire l'objet de requêtes ; les usages ne deviennent peu familiers que pour des fréquences *extrêmement* faibles (par exemple, « barrage de guitare », ou « ouverture de bridge », dont la fréquence est d'ailleurs impossible à évaluer précisément). Aucun des usages des mots-test utilisés dans la présente étude ne peut ainsi être qualifié de rare, et pourtant, nombre d'entre eux ont une fréquence de l'ordre de 1%. Nos tentatives de réplication de la technique de Schütze ont totalement échoué sur ces mots, ce qui nous a incité à développer une méthode beaucoup moins sensible à la fréquence, basée sur des propriétés particulières du graphe des cooccurrences.

3. « Petits mondes » lexicaux

On peut construire un graphe pour chaque mot à désambiguïser dans un corpus (ou mot-cible). Les nœuds de ce graphe sont les cooccurrents du mot-cible (par exemple dans une fenêtre de taille donnée, une phrase, un paragraphe, etc. — nous reviendrons plus loin sur les détails de construction). Une arête relie deux nœuds *A* et *B* chaque fois que les mots correspondants sont eux-mêmes en relation de cooccurrence. Ainsi, dans le graphe construit autour du mot-cible *barrage* (Figure 4), les nœuds correspondant à *production* et *électricité* seront interconnectés, car ils apparaissent ensemble dans des contextes tels que :

Outre la ***production*** *d'****électricité****, le* ***BARRAGE*** *permettra de réguler le cours du fleuve...*

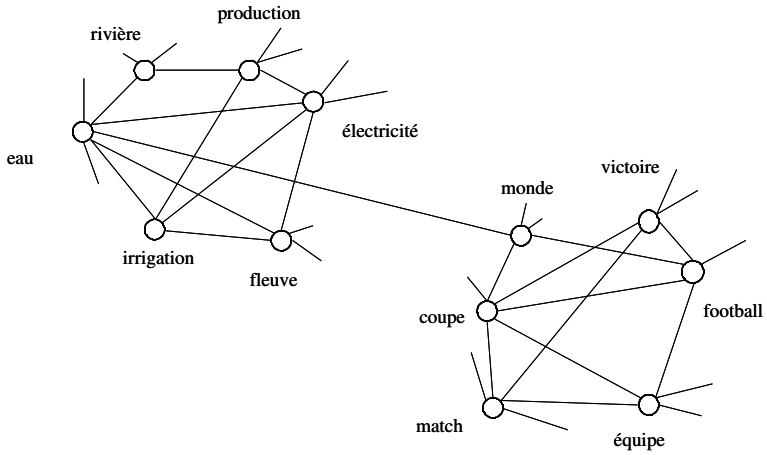

Figure 4. Graphe des cooccurrents du mot *barrage*

Nous montrons ci-après que ces graphes ont les propriétés des « petits mondes », mis en évidence par Watts & Strogatz (1998), et qui font l'objet d'un champ de recherche extrêmement important en théorie des graphes. Alors que l'essentiel des travaux en théorie des graphes avait porté sur des graphes réguliers, ou au contraire des graphes aléatoires, Watts & Strogatz (1998) et un courant croissant d'études ont montré que la plupart des graphes ou réseaux du monde réel ne relevaient ni de l'une ni de l'autre des catégories, et se situaient dans un état intermédiaire entre ordre et désordre.

1. Propriétés des graphes "petits mondes"

Watts & Strogatz (1998) définissent deux mesures qui permettent de caractériser les petits mondes : la *longueur de chemin caractéristique L,* et le *coefficient d'agglomération C.*

L est la moyenne des longueurs de plus court chemin entre deux nœuds du graphe. Soit $d_{\min}(i, j)$ la longueur du plus court chemin entre deux nœuds *i* et *j* et *N* le nombre total de nœuds :

$$L = \frac{1}{N} \sum_{i=1}^{N} d_{\min}(i, j) \tag{1}$$

Pour chaque nœud *i,* on peut définir un coefficient d'agglomération local C_i, qui est la proportion des connections $E(\Gamma(i))$ entre les voisins $\Gamma(i)$ de ce nœud. Pour un nœud *i* ayant 4 voisins, par exemple, le nombre maximal de connexions est de $\binom{|\Gamma(i)|}{2}$ = 6. Si 5 de ces connexions existent réellement, $C_i = 5/6 \sim 0.83$. Le coefficient global *C* est la moyenne des coefficients locaux :

$$C = \frac{1}{N} \sum_{i=1}^{N} \frac{|E(\Gamma(i))|}{\binom{|\Gamma(i)|}{2}} \tag{2}$$

Ce coefficient varie entre 0 pour un graphe totalement déconnecté) et 1 pour un graphe complet.

Dans le cas d'un graphe aléatoire de *N* nœuds et de degré moyen *k* (nombre moyen d'arêtes par nœud, c'est-à-dire *E/N, E* étant le nombre d'arêtes du graphe) :

$$L_{\text{rand}} \sim \log(N) / \log(k) \tag{3}$$

$$C_{\text{rand}} \sim 2\,k / N \tag{4}$$

Par exemple, un graphe aléatoire de 1000 nœuds et 10000 arêtes aura un degré moyen $k = 10$, une longueur de chemin caractéristique $L_{rand} \sim \log(1000)/\log(10) = 3$ et un coefficient d'agglomération $C_{rand} \sim 10/1000 = 0.01$.

Pour Watts & Strogatz (1998), ce qui caractérise un graphe de type « petit monde », ce sont les relations :

$$L \sim L_{rand} \quad (5)$$

$$C >> C_{rand} \quad (6)$$

La relation (5) signifie qu'à degré moyen constant, le nombre de nœuds peut croître de façon exponentielle, alors que la longueur de chemin caractéristique croîtra seulement de façon linéaire. Ceci explique le phénomène observé par Milgram (1967), qui est à l'origine du nom « petit monde » : n'importe quel individu de la planète est seulement à « six degrés de séparation » de n'importe quel autre dans le graphe des relations sociales, bien que le nombre d'habitants soit de plusieurs milliards.

L'équation (6) montre la différence entre un petit monde et un graphe aléatoire : dans un petit monde, on aura tendance à observer des « pelotes » correspondant à des groupes fortement interconnectés. Pour reprendre l'exemple des relations sociales, les amis d'un individu donné ont beaucoup plus de chances de se connaître entre eux que ne le prévoirait la simple répartition au hasard des arêtes dans le graphe.

A la suite de l'article de Watts & Strogatz (1998), les petits mondes ont fait l'objet de nombreux travaux, et cette structure a été découverte dans de très nombreux réseaux réels : Web, Internet, réseaux de mathématiciens ayant cosigné une article ou d'acteurs ayant joué ensemble dans un film, réseaux de distribution électrique, réseaux d'interaction de protéines, etc. (cf. Newman, 2003). La distribution des degrés des nœuds a également été examinée en détail. Alors que dans un graphe aléatoire, la probabilité $p(k)$ pour qu'un nœud soit de degré k diminue selon une loi exponentielle $p(k) = \beta\alpha^{-k}$ (loi de Poisson), les petits mondes observés respectent le plus souvent une loi de puissance (Barabási & Albert, 1999) :

$$p(k) = \beta\, k^{-\alpha} \quad (7)$$

avec α proche de l'unité.

Les graphes respectant cette propriété sont dits indépendants de l'échelle (*scale-free*)[4]. Dans un graphe de ce type, on observe que la plupart des nœuds sont peu connectés, tandis qu'un très petit nombre de nœuds (*hubs*) sont au contraire connectés à un très grand nombre d'autres. La suppression de ces derniers peut entraîner des dommages considérables dans le réseau. C'est typiquement le cas de l'Internet.

2. Construction des graphes de cooccurrences

Nous avons choisi 10 noms très polysémiques parmi ceux qui avaient posé de grandes difficultés à des annotateurs humains dans Véronis (1998) (Tableau 1). Un sous-corpus de pages Web a été constitué pour chacun de ces mots, à l'aide du méta-moteur Copernic Agent[5], en interrogeant tout d'abord la forme singulier, puis la forme pluriel. Les pages obtenues ont été filtrées de façon à éliminer celles qui ne contenaient pas le mot cherché (erreurs du type « Page not found », par exemple), ainsi que les doublons.

Les paragraphes contenant chaque mot-cible ont été extraits et étiquetés à l'aide du logiciel Cordial Analyseur[6], augmenté d'un certain nombre de programmes de post-traitement. Seuls ont été retenus les noms et les adjectifs. Dans un premier temps nous avions retenu aussi les verbes, mais il s'est finalement avéré que ceux-ci dégradent notablement les performances, trop de verbes ayant des

[4] A l'inverse, les graphes aléatoires ont une échelle, le degré moyen *k,* qui est le pic de la distribution des degrés.

[5] http://www.copernic.com

[6] Développé par Synapse Développement : http://www.synapse-fr.com

usages généraux (par exemple, *commencer, pouvoir,* etc.) — ceci n'est qu'une solution temporaire, et constitue un point de recherches ultérieures. Les paragraphes ont filtrés pour éliminer les mots-outils (déterminants, prépositions, etc.), ainsi qu'un certain nombre de mots généraux appartenant à un anti-dictionnaire (*stoplist*) et tout particulièrement, étant donné notre application, ceux liés au Web lui-même (*menu, accueil, lien, http,* etc.)[7]. Les mots qui ont moins de 10 occurrences dans la totalité du sous-corpus ont également été filtrés. Finalement, les contextes contenant moins de 4 mots après filtrage ont été éliminés.

A partir des contextes filtrés, nous construisons la matrice des cooccurrences : deux mots qui apparaissent dans le même paragraphe sont considérés comme étant en cooccurrence[8]. Seules les cooccurrences de fréquence ≥ 5 ont été retenues.

Le Tableau 1 donne les caractéristiques quantitatives du sous-corpus recueilli pour chaque mot, ainsi que du graphe de cooccurrences qu'il a permis de construire.

Mot-cible	**Traduction**	**Pages**		**Contextes**	
		Brutes	Utiles	Bruts	Utiles
BARRAGE	*dam, blockade, barrage...*	1702	1372	7256	6924
DETENTION	*detention, possession, holding, custody…*	2112	1270	8902	8728
FORMATION	*training, formation*	5974	1590	5248	4885
LANCEMENT	*launching, starting up, throwing...*	2828	1231	3307	3174
ORGANE	*organ, instrument, medium, representative…*	2786	994	2953	2849
PASSAGE	*passage, way, crossing, transition, coming by…*	3512	1046	4210	3894
RESTAURATION	*restoration, rehabilitation, catering, food industry…*	5327	1227	3522	3287
SOLUTION	*solution, answer*	6287	896	2085	1915
STATION	*station, halt, site…*	7916	1093	3837	3671
VOL	*flight, gliding, theft, robbery…*	5237	818	3001	2579

Tableau 1. Mots-cibles et caractéristiques quantitatives des sous-corpus

3. Pondération

Nous affectons à chaque arête un poids d'autant plus faible que les mots sont fréquemment associés :

$$w_{A,B} = 1 - \max[\, p\,(A \mid B), p(B \mid A)\,] \qquad (8)$$

où $p(A \mid B)$ est la probabilité conditionnelle d'observer A dans un contexte donné sachant que ce contexte contient B, et inversement, $p(B \mid A)$ celle d'observer B dans un contexte donné sachant que ce contexte contient A. Ces probabilités sont estimées à partir des fréquences :

$$p(A \mid B) = f_{A.B} / f_B \quad \text{et} \quad p(B \mid A) = f_{A.B} / f_A \qquad (9)$$

[7] La qualité de la lemmatisation et celle du filtrage sont extrêmement importantes. Si la méthode est robuste dans son ensemble, des erreurs systématiques de lemmatisation portant sur les *hubs* du graphe peuvent avoir des résultats désastreux, tout autant que la présence de mots non filtrés tels que *menu* ou *accueil* qui créent artificiellement des *hubs* non liés aux thématiques du sous-corpus concerné. La précision de l'étiquetage morphosyntaxique que nous obtenons sur les catégories majeures (nom, adjectif, verbe) est de l'ordre de 99% (nous sommes aidés par le fait que les principales difficultés concernent les distinctions entre catégories mineures : préposition/adverbe, etc.).

[8] D'autres pistes pourraient être explorées, comme l'utilisation d'une fenêtre de taille fixe à travers le texte. Cependant, le choix du paragraphe comme unité contextuelle est intéressant dans la perspective d'une vraie application, car il permet de construire une seule matrice de cooccurrences pour l'ensemble du corpus, valable pour tous les mots à désambiguïser, économisant ainsi un temps de traitement considérable.

Nous prendrons à titre d'illustration les cooccurrences *eau - ouvrage* et *eau - potable.* Le Tableau 2 donne le nombre de contextes dans lesquels ces couples de mots apparaissent ensemble ou l'un sans l'autre dans le sous-corpus *barrage*. On voit que toutes les occurrences du mot *potable* apparaissent conjointement avec le mot *eau,* alors que c'est le cas seulement d'une partie des occurrences du mot *ouvrage.*

	EAU	**~EAU**	**Total**
OUVRAGE	183	296	479
~OUVRAGE	874	5556	6430
Total	1057	5852	6909

	EAU	**~EAU**	**Total**
POTABLE	63	0	63
~POTABLE	994	5852	6846
Total	1057	5852	6909

Tableau 2. Cooccurrences *eau-ouvrage* et *eau-potable*

On a :

$p(eau \mid ouvrage) = 183/479 = 0{,}38$ $\quad p(ouvrage \mid eau) = 183/1057 = 0{,}17$ $\quad w = 1 - 0{,}38 = 0{,}62$
$p(eau \mid potable) = 63/63 = 1$ $\quad p(potable \mid eau) = 63/1057 = 0{,}06$ $\quad w = 1 - 1 = 0$

La mesure reflète donc la plus ou moins grande « distance[9] » sémantique entre mots : lorsqu'elle vaut 0, les mots sont toujours associés (jusqu'à concurrence de la fréquence maximale possible, déterminée par le moins fréquent) ; lorsqu'elle vaut 1, les mots ne sont jamais associés.

Les arêtes qui dépassent un poids de 0.9 ont été arbitrairement éliminées. Ce seuillage est très important, car il permet que seules les arêtes correspondant à une association forte soient incluses dans le graphe. En son absence, le graphe aurait tendance à devenir totalement connecté lorsque le corpus devient grand, à cause de la présence de plus en plus probable de cooccurrences accidentelles entre n'importe quelle paire de mots.

La pondération des arêtes nous permet de définir un coefficient d'agglomération pondéré $C^{'}$:

$$C^{'} = \frac{1}{N}\sum_{i=1}^{N} \frac{\sum_{j=1}^{|E(\Gamma(i))|}(1 - w_{ij})}{\binom{|\Gamma(i)|}{2}} \qquad (10)$$

Ce coefficient est un peu plus fin que le coefficient original de Watts & Strogatz (1998) : au lieu de tenir compte simplement de la présence ou de l'absence d'une arête, il tient compte également de leurs poids respectifs.

4. Propriétés des graphes de cooccurrences

Après ces différentes opérations, les graphes obtenus ont les caractéristiques listées dans le Tableau 3.

Mot	***N***	***E***	***k***	***C***	***L***	C_{rand}	L_{rand}
BARRAGE	1203	6138	5,1	0,47	3,5	0,008	4,4
DETENTION	1418	19007	13,4	0,55	3,3	0,019	2,8
FORMATION	542	1531	2,8	0,44	3,5	0,010	6,1
LANCEMENT	617	2521	4,1	0,52	3,6	0,013	4,6
ORGANE	531	1997	3,8	0,44	4,0	0,014	4,7
PASSAGE	797	2916	3,7	0,47	4,5	0,009	5,2
RESTAURATION	512	1398	2,7	0,46	4,0	0,011	6,2
SOLUTION	253	1704	6,7	0,57	2,1	0,053	2,9
STATION	487	971	2,0	0,43	3,7	0,008	9,0
VOL	259	719	2,8	0,48	2,7	0,021	5,4

[9]Il ne s'agit pas d'une *distance* au sens mathématique du terme, mais d'une *dissimilarité,* l'inégalité triangulaire n'étant pas respectée.

Tableau 3. Caractéristiques des graphes

On voit que les relations (5) et (6) sont toutes deux respectées, impliquant que les graphes de cooccurrences sont de type petit monde. De plus, la relation entre $p(k)$ et k est approximativement régie par une loi de puissance, comme le montre la Figure 5 dans le cas du mot *barrage*. Les réseaux de cooccurrences sont indépendants de l'échelle, et on y observe donc la présence d'un petit nombre de hubs fortement connectés, en combinaison avec un grand nombre de nœuds peu connectés.

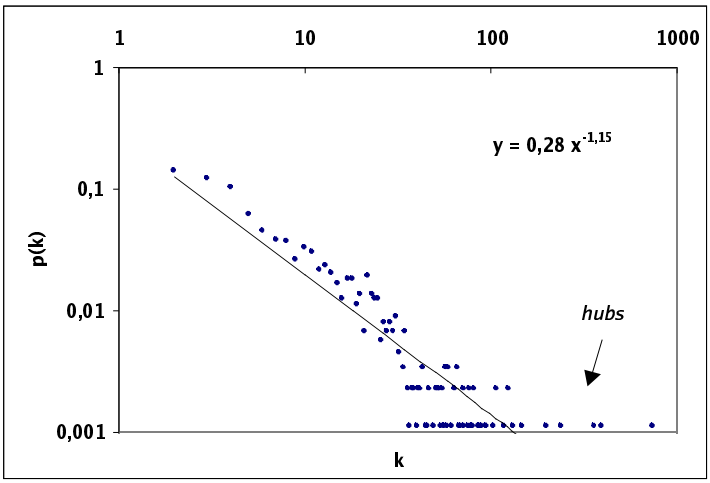

Figure 5. Loi de puissance sur les degrés pour le mot *barrage*

On observe enfin que degrés et fréquence des mots sont très fortement corrélés, de façon presque linéaire (Figure 6). Nous exploiterons cette propriété pour simplifier certains calculs.

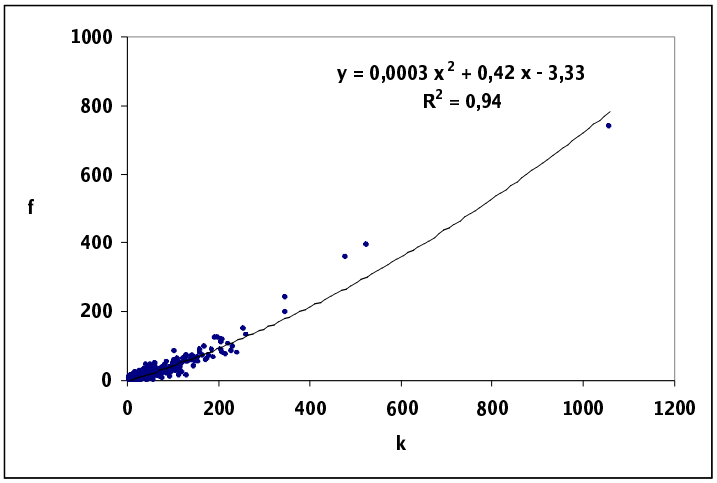

Figure 6. Corrélation entre degré et fréquence (*barrage*)

4. Détection des composantes de forte densité

L'hypothèse de base qui sous-tend notre méthode est que les différents usages du mot-cible constituent des « pelotes » fortement interconnectées dans le petit monde des cooccurrences, ou, en termes de théorie des graphes, des *composantes de haute densité.* En effet, *barrage* (dans l'usage « barrage hydraulique ») doit être en cooccurrence fréquente avec *eau, ouvrage, rivière, crue, irrigation, production, électricité,* etc., et ces mots eux-mêmes ont toutes les chances d'être interconnectés (Figure 4). De même, dans l'usage « match de barrage », *barrage* doit être en cooccurrence fréquente avec *match, équipe, coupe, monde, football, victoire,* etc., ces termes eux-mêmes étant fortement interconnectés. Etant donné la complexité du langage (et en particulier le fait

que les mots entrant dans les cooccurrences sont eux-mêmes ambigus), il existe aussi des connexions entre les composantes, ce qui interdit l'utilisation d'algorithmes de détection de composantes fortement connexes ou de cliques, mais ces interconnections entre les composantes doivent être peu nombreuses et leur poids élevé.

Détecter les différents usages d'un mot revient donc à isoler des *composantes de forte densité* dans le graphe des cooccurrences. La plupart des techniques exactes de partitionnement de graphes sont malheureusement NP-difficiles, et l'on ne peut (étant donné que les graphes obtenus ont plusieurs milliers de nœuds et d'arêtes) qu'utiliser des méthodes approximatives, basées sur des heuristiques. La recherche sur la détection de composantes de forte densité est un domaine particulièrement actif, qui intervient notamment dans les secteurs de la détection de « communautés » ou de « sources autorisées » sur le Web, ou la parallélisation des calculs. Malheureusement, les techniques développées dans ces secteurs ne sont pas directement exploitables, étant donné que les heuristiques dépendent des applications et des propriétés particulières des graphes analysés.

Nous exploiterons ici les propriétés des petits mondes décrites plus haut, ainsi que l'indépendance d'échelle que nous avons mise en évidence.

L'algorithme se décompose en deux étapes : dans un premier temps nous détectons un certain nombre de hubs particuliers qui servent de « racines » aux différentes composantes ; dans un deuxième temps, nous listons les nœuds qui appartiennent à chacune des composantes. La première étape suffit à énumérer la liste des usages du mot-cible dans le corpus ; la deuxième est nécessaire pour la désambiguïsation et pour la visualisation.

1. Détection des hub racines

Nous partons de la remarque que dans chacune des composantes de forte densité, l'un des nœuds a un degré plus élevé que les autres. Nous l'appellerons hub racine de la composante. Par exemple, pour l'usage le plus fréquent de *barrage* (barrage hydraulique), le hub racine est le mot *eau.* Il est facile à repérer, puisqu'il est le hub de plus fort degré du graphe (et c'est aussi le mot le plus fréquent).

Il s'agit ensuite de repérer le hub racine de la composante suivante. La structure du graphe, constitué de « pelotes » fortement connectées de façon interne, mais peu connectées les unes aux autres, nous permet d'appliquer une stratégie simple : en supprimant du graphe le hub racine que l'on vient d'isoler *ainsi que tous ses voisins,* il y a de grandes chances qu'on élimine la quasi-totalité de la première composante de forte densité. En effet, d'après l'organisation même des petits mondes, si un mot de degré raisonnablement important fait partie de cette première composante, il est aussi relié de façon multiple aux nœuds qui la composent, et il est très hautement probable qu'il soit connecté aussi au hub racine. A contrario, s'il ne l'est pas, on peut être raisonnablement sûr qu'il fait partie d'une autre composante.

Cette stratégie supprime manifestement des nœuds qui ne font pas partie de la première composante. Ainsi dans l'exemple de la Figure 4, le nœud *monde* sera supprimé même s'il fait partie de la composante « match de barrage ». L'hypothèse qui est faite ici est que ces liens intercomposantes étant peu nombreux, il restera suffisamment de nœuds propres à cette deuxième composante (et notamment son hub racine) pour permettre sa détection.

L'algorithme continue de façon itérative. Le hub candidat suivant est le nœud *routier,* lui même lié à *véhicule, camion,* etc. Il est supprimé, ainsi que ses voisins, et ainsi de suite, jusqu'à épuisement des nœuds du graphe (Figure 7).

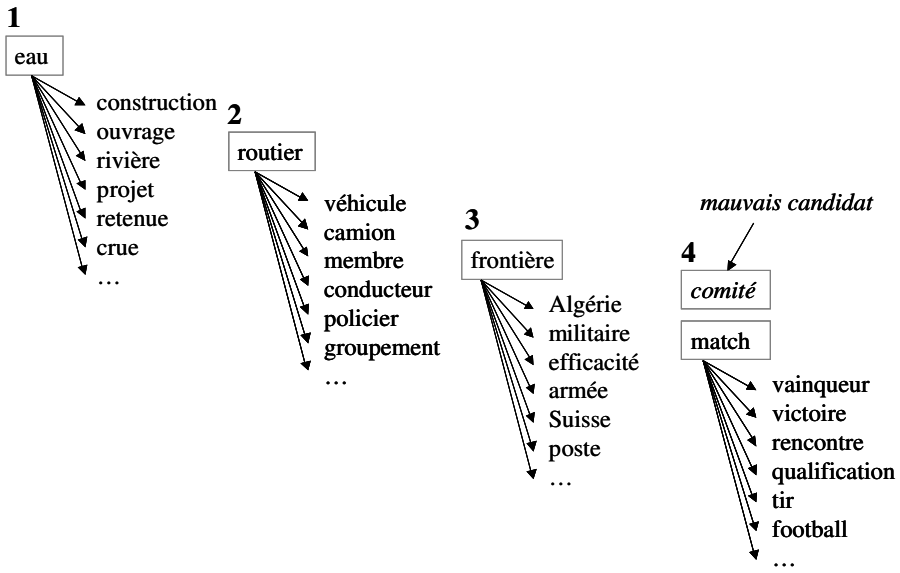

Figure 7. Suppression successive des voisins

Lorsqu'on atteint les composantes les plus petites du graphe, le risque devient important que trop de nœuds aient été supprimés, et quelques heuristiques correctives sont nécessaires pour s'assurer que le prochain nœud examiné est bien un bon candidat au statut de hub racine. Nous vérifions que :

1. il possède un nombre de voisins propres supérieur ou égal à un seuil fixé (que nous avons établi expérimentalement à 6) ;
2. que son coefficient d'agglomération pondéré est suffisant pour qu'il soit réellement hub racine d'une pelote.

En fait, dans un souci de rapidité, nous utilisons une approximation grossière pour le critère 2, qui s'avère suffisante : nous imposons simplement que la moyenne des poids entre le nœud candidat et ses 6 voisins les plus fréquents soit inférieure à un seuil donné (établi expérimentalement à 0.8). De même, au lieu de parcourir les nœuds par ordre de degré décroissant, ce qui implique un calcul au niveau de chaque nœud, nous les parcourons par ordre de *fréquence* décroissante, cette information étant déjà calculée lors du prétraitement des contextes. Les deux étant fortement corrélés (voir plus haut), le résultat est quasiment identique.

A l'issue de cette première étape, on dispose des hubs racines de chacune des composantes. Combinés à leurs voisins propres les plus fréquents, ils permettent de caractériser chacun des usages.

Pour *barrage,* on a par exemple quatre composantes, qui sont caractérisées de la façon suivante :

EAU	*construction ouvrage rivière projet retenue crue*
ROUTIER	*véhicule camion membre conducteur policier groupement*
FRONTIERE	*Algérie militaire efficacité armée Suisse poste*
MATCH	*vainqueur victoire rencontre qualification tir football*

Cette information peut être suffisante, par exemple s'il s'agit de la présenter à un utilisateur en lui demandant de préciser sa requête. Elle ne permet pas toutefois de délimiter la composition exacte de la composante (ceci fait l'objet de la deuxième étape de l'algorithme).

De façon plus formelle, l'algorithme peut s'écrire comme suit :

```
HubsRacines(G, Freq) {
    G : graphe de cooccurrences
    Freq : tableau des fréquences de chaque nœud de G

    V ← tableau des nœuds de G triés par fréquence décroissante
    H ← Ø

    tant que V ≠ Ø et Freq(V[0]) > seuil {
        v ← V[0]
        si BonCandidat(v)
        alors {
            H ← H ∪ v
            V ← V – (v ∪ Γ(v))
        }
    retourner H
}
```

L'algorithme est très rapide puisqu'une fois les nœuds triés par fréquence décroissante (ce qui intervient en fait dans l'étape de préparation du corpus), il fonctionne en $O(N)$, N étant le nombre de nœuds du graphe (le nombre d'opérations de suppression de voisins est au maximum égal à N).

2. Délimitation des composantes

Délimiter les composantes de forte densité revient à rattacher chaque nœud au hub racine qui lui est le plus proche. Comme, étant donné la structure de « petit monde », tous les nœuds sont proches les uns des autres en termes de nombre d'arêtes à traverser, il est intéressant d'utiliser la pondération des arêtes du graphe : la distance entre deux nœuds s'évaluera par la somme minimale des poids des arêtes sur les chemins qui les relient. Après avoir ajouté le mot-cible (qui ne fait partie du graphe des cooccurrences — ici, *barrage*), nous calculons un arbre de couverture minimal (*minimum spanning tree* ou MST) sur le graphe, en prenant le mot-cible pour racine, et en imposant que son premier niveau soit constitué des hubs racines précédemment dégagés[10]. Les composantes correspondent aux branches principales de l'arbre.

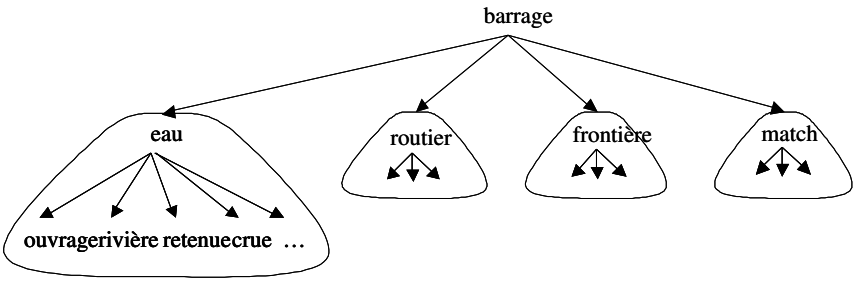

Tableau 4. Arbre de couverture minimal et composantes de forte densité

[10] Un arbre de couverture minimal est un arbre qui passe par tous les nœuds du graphe initial, tout en minimisant la somme des poids des arêtes ; divers algorithmes standards sont disponibles pour calculer des arbres de couverture minimal de façon efficace

L'algorithme est le suivant :

```
Composantes(G, H, t) {
    G : graphe de cooccurrences
    H : ensemble des hubs racines
    t : mot-cible

    G' ← G ∪ t
    pour chaque h dans H {
        ajouter arête <t, h> de poids 0 à G'
    }

    T ← MST(G', t)

    retourner T
}
```

La complexité de l'algorithme se réduit à celle du calcul de l'arbre de couverture minimal peut être calculé efficacement par l'algorithme de Kruskal (1956), bien adapté aux graphes clairsemés (*sparse graphs*). Sa complexité dans le pire des cas est $O(E \lg E)$, E étant le nombre d'arêtes du graphe. Toutefois, on sait que l'algorithme de Kruskal se comporte de façon quasi-linéaire dans la plupart des cas concrets.

5. Désambiguïsation

L'arbre de couverture minimal permet de construire aisément un désambiguïsateur, qui pourra être appliqué à l'étiquetage des occurrences du mot-cible dans le corpus. A chaque nœud v de l'arbre est attribué un vecteur de scores **s** ayant autant de dimensions que de composantes :

$$\mathbf{s}_i = \frac{1}{1 + d\left(h_i, v\right)} \quad \text{si } v \text{ appartient à la composante } i \tag{11}$$

$$\mathbf{s}_i = 0 \text{ sinon.} \tag{12}$$

En (11), $d(h_i, v)$ est la distance entre le hub racine h_i et le nœud v dans l'arbre.

La formule (11) permet d'attribuer le score 1 aux hubs racines, qui sont à une distance de 0 d'eux-mêmes. Le score se rapproche progressivement de 0 au fur et à mesure que les nœuds s'éloignent de leur hub racine. Par exemple, *pluie* appartient à la composante *EAU* et $d(eau, pluie)$ = 0.82. Son vecteur de score est égal à (0.55 0 0 0). De même, *saison* appartient à la composante *MATCH* et $d(match, saison)$ = 1.54. Son vecteur de score est égal à (0 0 0 0.39).

Face à un contexte donné, les vecteurs de scores de tous les mots de ce contexte sont additionnés, et la composante qui reçoit le poids le plus fort est choisie.

Par exemple, le contexte suivant :

*Le **barrage** recueille l'eau à la saison des pluies.*

fait l'objet du calcul donné par le Tableau 5. On voit que la composante qui reçoit le plus fort score total est la composante *EAU* (1.55), suivie par la composante *MATCH* (0.39).

	EAU	**ROUTIER**	**FRONTIERE**	**MATCH**
S_{eau}	1.00	0.00	0.00	0.00
S_{saison}	0.00	0.00	0.00	0.39
S_{pluie}	0.55	0.00	0.00	0.00
Total	**1.55**	**0.00**	**0.00**	**0.39**

Tableau 5. Calcul des scores

Un coefficient de fiabilité peut également être calculé, à partir de la différence Δ entre le meilleur score et le score immédiatement inférieur. De façon à obtenir une valeur entre 0 et 1, le coefficient de fiabilité se calcule de la façon suivante :

$$= 1 \quad - \quad \frac{1}{1+\Delta}$$

Dans l'exemple précédent, le coefficient de fiabilité ρ est de 1 - (1 /(1 + 1.55 - 0.39) = 0.54.

Il est intéressant de se pencher plus en détail sur ce que fait exactement l'algorithme du point de vue linguistique. Nous partirons de l'exemple réel de la Figure 8. On voit, sur la partie gauche de la figure, que les mots *eau, navigable, force, cours, moteur* sont fortement interconnectés (5 connections sur les 6 possibles). Or ces connections ne sont pas toutes de même nature. Les relations *eau-cours, cours-navigable, eau- force*, *force-moteur*, sont des relations primaires, qui apparaissent d'ailleurs dans des expressions du type *cours d'eau, cours navigable*, *force de l'eau, force motrice.* Les autres relations sont des relations *secondaires,* qui apparaissent par transitivité, par un principe du type « les amis de mes amis deviennent aussi mes amis » : il n'y a pas de relation particulière entre *eau* et *moteur,* si ce n'est par l'intermédiaire de la *force de l'eau,* qui est *motrice*. Le calcul de l'arbre de couverture minimal permet de « désagglomérer » le graphe, en mettant en évidence les relations primaires entre mots. L'algorithme « défait » donc en quelque sorte le « petit monde » lexical, en nous montrant les relations préférentielles : parmi les 4 voisins de *eau* pris en exemple, seuls deux restent ses voisins dans l'arbre résultant.

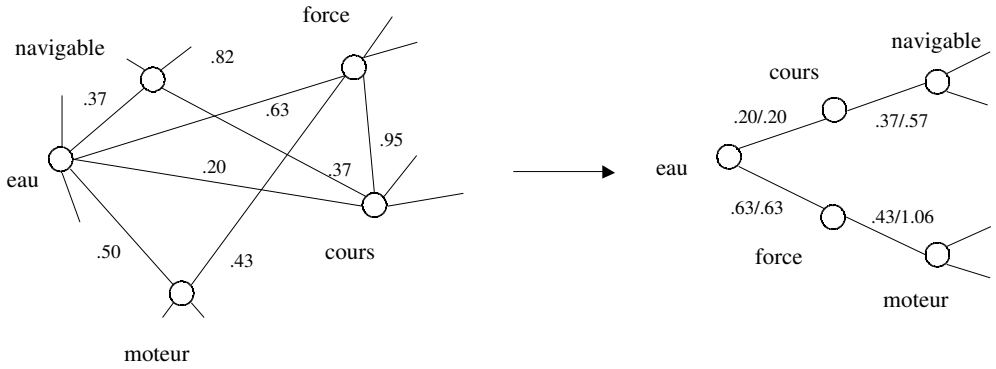

Figure 8. « Développement » du graphe

6. Visualisation et navigation

La visualisation de graphes de grandes dimensions est un problème difficile à la fois du point de vue informatique (surtout si l'on vise un tracé suffisamment rapide pour permettre navigation et manipulation en temps réel, la plupart des tracés étant *NP*-difficiles : Brandeburg, 1988), et ergonomique (car la quantité d'information risque de rendre la représentation illisible). Etant donné l'importance croissante des réseaux de grande dimension dans divers secteurs d'application (dont le Web et l'Internet), de multiples recherches sont en cours et de nombreuses solutions adaptées à des problèmes particuliers ont été proposées (cf. Di Battista *et al.,* 1999).

La méthode récemment proposée par Munzner (2000) semble particulièrement adaptée à notre problème, puisqu'elle s'applique à une classe de graphes que Munzner appelle « quasi-hiérarchique », c'est-à-dire qui possèdent de façon naturelle une structure sous-jacente en forme d'arbre. Un site Web est un exemple de graphe quasi-hiérarchique : bien que des hyperliens puissent apparaître de façon libre entre n'importe quelle paire de documents, la structure des répertoires qui les contiennent constitue une ossature (*backbone*) en forme d'arbre. Munzner remarque que cette ossature hiérarchique a le plus souvent une pertinence éditoriale, et constitue donc un point de départ intuitif pour la visualisation et la navigation. L'affichage d'arbres étant rapide, cela permet à Munzner le

développement d'un système d'affichage qui traite en temps réel des graphes de plusieurs dizaines des milliers d'arêtes.

Les arbres de très grande taille sont également difficiles à représenter, car le nombre de nœuds croît de façon exponentielle avec la profondeur, tandis que l'espace euclidien croît seulement de façon polynomiale. On se retrouve donc inévitablement à partir d'une certaine taille avec une information trop dense pour l'affichage, deux nœuds distincts ne pouvant plus être discriminés à l'écran. Munzner (2000) réalise un tracé dans un espace hyperbolique, car un tel espace a justement la propriété de croître de façon exponentielle, ce qui permet la représentation de la totalité de l'arbre avec suffisamment de place pour tous les nœuds. Cet espace est ensuite projeté sur un disque plan (Figure 9), ce qui produit pour l'utilisateur un effet « fish-eye », dans lequel les détails situés au centre du disque apparaissent en gros plan, puis donnent l'impression de se rétrécir à l'infini au fur et à mesure qu'on se rapproche des bords du disque.

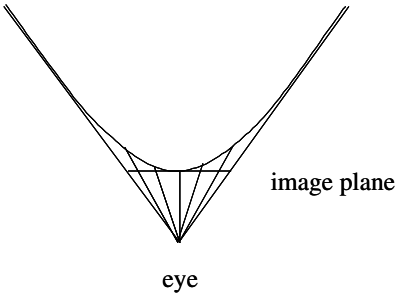

Figure 9. Représentation hyperbolique (projection sur un disque plan ; d'après Munzner, 2000)

Nous disposons justement dans notre cas d'une ossature en forme d'arbre, constituée par l'arbre de couverture minimal calculé à la section précédente. Nous avons d'ailleurs vu qu'il faisait apparaître les relations « primaires » du graphe. Il semble donc que cet arbre soit un bon candidat du point de vue cognitif pour la lecture par l'utilisateur. Nous avons donc utilisé la librairie graphique *H3Viewer* fournie par Munzner[11], et nous avons développé quelques heuristiques d'élagage permettant une simplification de l'arbre. En effet, appliqué de façon stricte, l'algorithme affecte obligatoirement chaque nœud du graphe à une des composantes. Or, il ne faut pas oublier que notre procédure est une heuristique, et non une technique de partitionnement exacte. Les rattachements deviennent nécessairement douteux dans les parties les plus profondes de l'arbre. De plus, leur lien aux thématiques principales du mot-cible devient de plus en plus lointain.

Nous avons appliqué une stratégie qui semble satisfaisante, et qui consiste à :

1. supprimer du graphe les arêtes dont les extrémités ont des fréquences trop différentes (nous avons fixé un rapport maximal de 1 à 8) ;
2. ajouter un biais favorisant les 6 voisins les plus fréquents de chaque hub racine (leur poids est divisé par 4) ;
3. couper les branches de l'arbre résultant au-delà d'une profondeur donnée (7).

La Figure 10 montre la vue principale (sommet de l'arbre) pour le mot *barrage* (les hubs racines apparaissent en rose).

L'utilisateur peut naviguer de thème en thème à l'intérieur de la représentation hyperbolique à l'aide de la souris : un clic sur un nœud avec le bouton gauche permet de centrer la représentation autour de ce nœud, un glisser avec le bouton gauche appuyé permet de déplacer un nœud et de changer le contexte, un glisser avec le bouton droit permet une rotation de l'arbre. La Figure 11 montre la vue obtenue en cliquant sur le nœud *match.* Une addition, en cours d'implémentation, permettra de visualiser à la demande les contextes du corpus les plus proches de chaque nœud de l'arbre.

[11] http://graphics.stanford.edu/~munzner/h3/

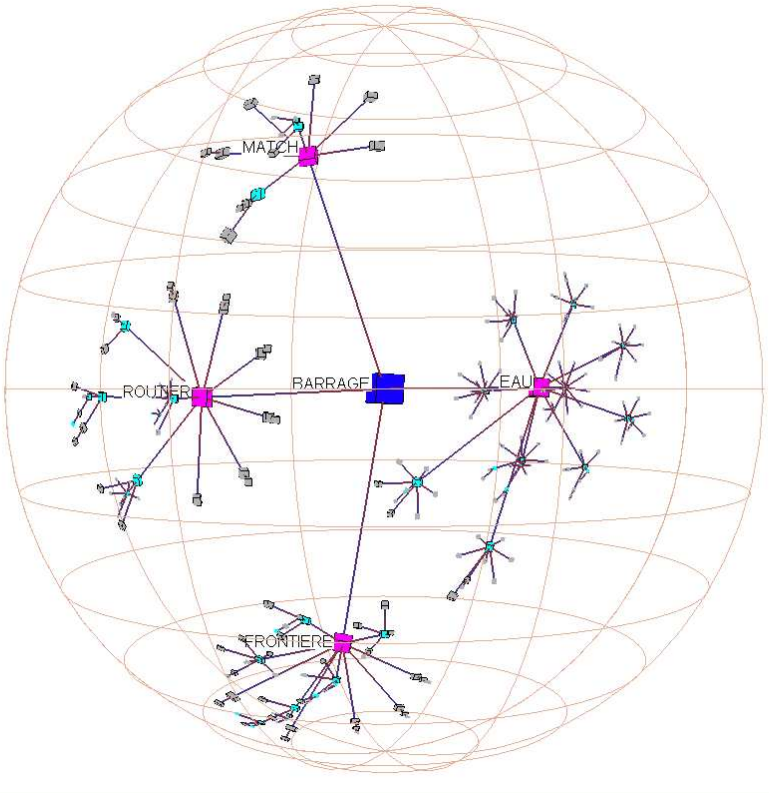

Figure 10. *Barrage* : vue principale

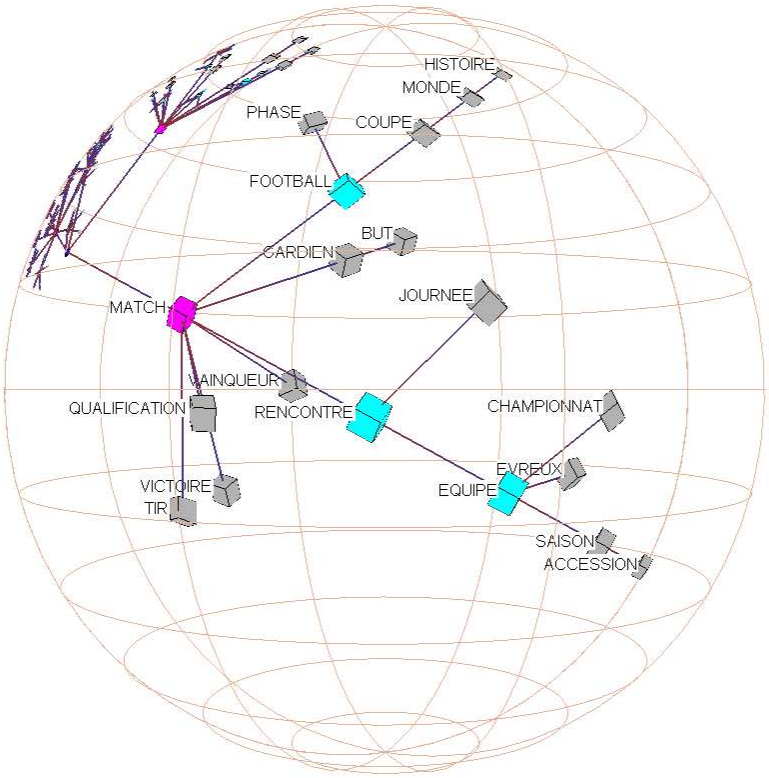

Figure 11. *Barrage* : vue recentrée sur le hub racine *match*

La navigation à travers le graphe permet également d'explorer les thématiques secondaires à l'intérieur d'une composante donnée. Ainsi, la Figure 12 illustre une sous-thématique à l'intérieur de l'usage « barrage hydraulique », obtenue en cliquant sur *construction,* puis sur *coût*.

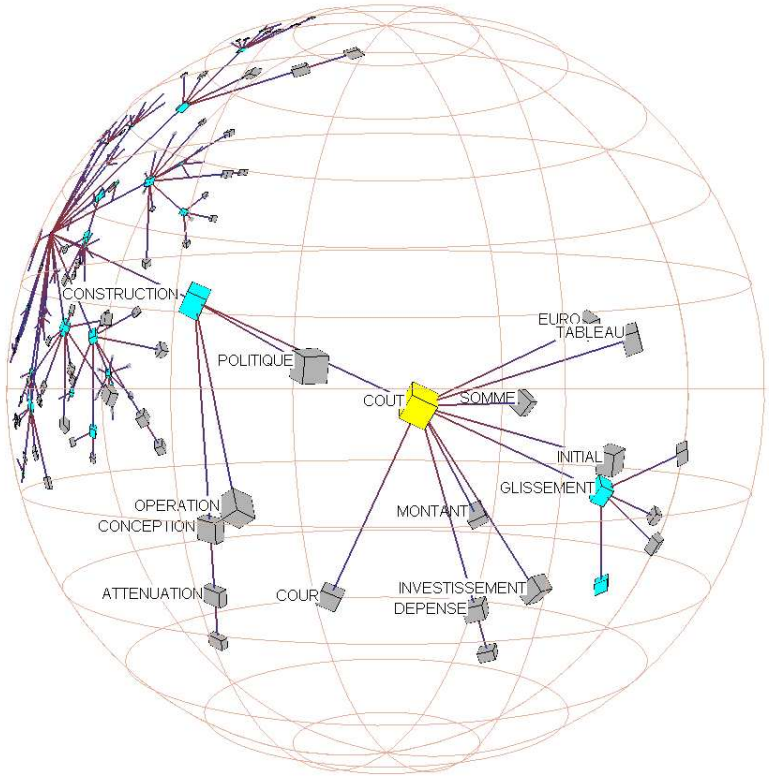

Figure 12. *Barrage :* thématique *construction* → *coût*

Enfin, le programme permet d'afficher la totalité des arêtes du graphe, c'est-à-dire également les liens transversaux, et non seulement ceux de l'arbre ossature. Cette représentation permet de juger de la répartition entre liens intra-composante et liens inter-composantes. La Figure 13 montre par exemple les liens transversaux pour le mot-cible *barrage,* et l'on voit que les arêtes inter-composantes sont peu nombreuses : les usages sont bien distincts. A l'inverse, la Figure 14, concernant le mot *vol,* montre d'importantes liaisons transversales entre les composantes *LIBRE* et *AVION*. En naviguant dans le graphe et examinant plus en détail les raisons de ces relations, on s'aperçoit que les deux composantes partagent une sous-thématique commune importante, celles des vacances (loisirs, soleil, etc.). A l'inverse, la composante *vol à voile* est peu liée à la thématique des vacances (du moins dans le corpus) : les pages sont surtout techniques.

HyperLex semble donc pourvoir fournir un outil de navigation lexicale et thématique utile. Il reste à savoir si l'utilisation par un public général en est réaliste, mais il semblerait qu'il puisse fournir un outil d'exploration intéressant pour terminologues et lexicographes.

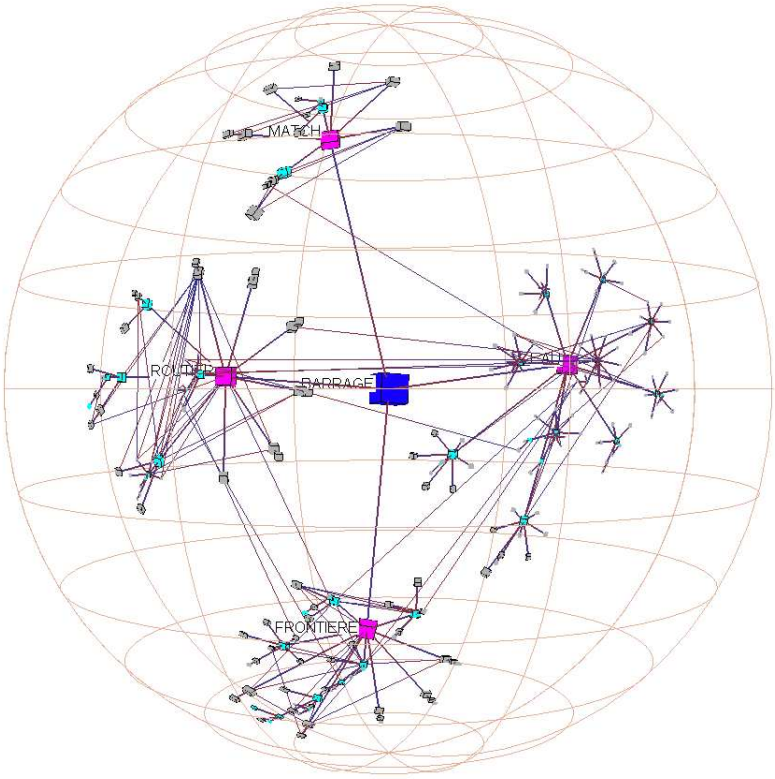

Figure 13. *Barrage* : représentation complète du graphe

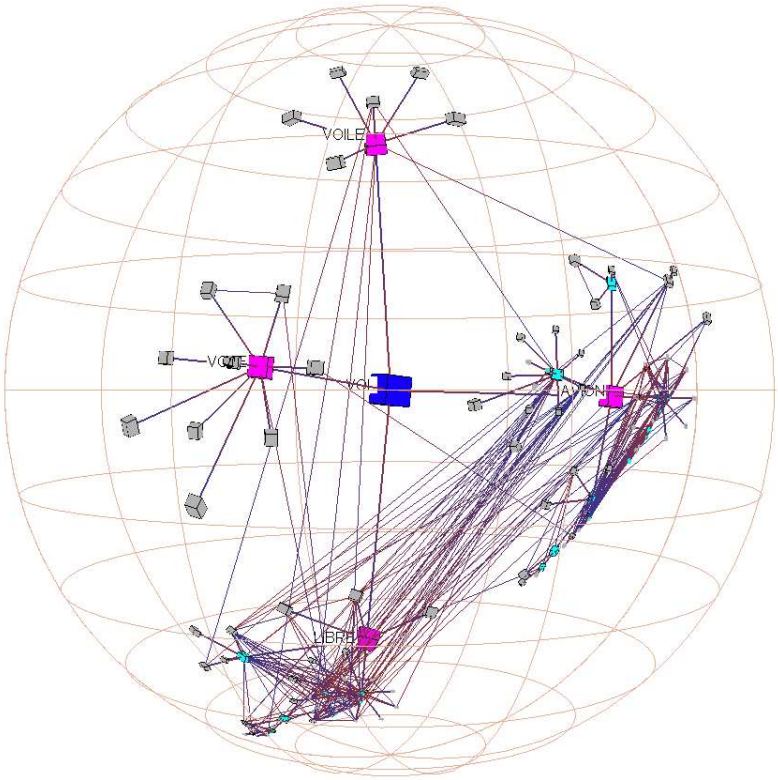

Figure 14. *Vol* : liens entre composantes (*vol libre* ↔ *avion*)

7. Evaluation

Nous avons évalué les résultats de l'algorithme *HyperLex* sur le corpus de pages Web et avec la liste des dix mots-test décrits plus haut.

Tout d'abord, nous avons vérifié si l'algorithme extrait correctement la plupart des usages dans le corpus, indépendamment de tout étiquetage : en effet, cette sous-tâche est intéressante en soi (par exemple pour proposer à l'utilisateur un raffinement de sa requête).

Nous avons ensuite évalué la qualité de l'étiquetage de contextes tirés au hasard dans le corpus, fournissant ainsi des mesures de rappel et précision classiques en désambiguïsation lexicale. Toutefois, ces mesures ne permettent pas de juger correctement l'efficacité de l'algorithme sur les usages peu fréquents : elle reflètent surtout le comportement sur l'usage majoritaire. Nous proposons donc une troisième mesure, beaucoup plus sévère : la précision sur les 25 meilleurs contextes retournés par l'algorithme pour *chacun* des usages discriminés.

1. Liste des usages

Le Tableau 7 montre la liste des usages extraits par *HyperLex* sur l'ensemble du corpus. 50 usages ont été repérés au total, soit une moyenne de 5 par mot-cible. Afin d'évaluer la complétude et la pertinence de cette liste, nous avons tiré au hasard 100 contextes pour chacun des mots-test, soit 1000 contextes au total.

Le système fait deux sortes d'omissions. Un certain nombre d'entre elles concernent des usages généraux des mots-cibles, c'est-à-dire des usages qui ne sont pas liés à une thématique particulière. Ainsi pour le mot *barrage,* l'usage *faire barrage à* n'est pas repéré. Nous n'en trouvons que 4 exemples dans l'échantillon de contextes testés, c'est-à-dire à peu près l'ordre de grandeur de l'usage *match de barrage* : il aurait donc pu quantitativement être repéré. Il se trouve toutefois que cette expression est une expression générale de la langue, qui se retrouve dans les contextes les plus divers : *faire barrage à l'extrême-droite, à un projet, à quelqu'un,* etc. Un autre exemple caractéristique est celui du mot *solution,* pour lequel l'algorithme ne détecte pas le sens général « solution à un problème », assez fréquent (16 occurrences), mais qui peut apparaître dans n'importe quelle thématique.

Au total 8 usages généraux sont manquants pour la totalité des 10 mots-cibles, dont 3 seulement de fréquence supérieure à 5 (Tableau 6). Ces omissions ne sont pas très gênantes, car elles ne portent pas sur des usages à caractère thématique, et ne peuvent, selon nous, guère faire l'objet de requêtes. Elle ne concernent d'ailleurs que 6.6% des contextes.

	Tous		**Fréquence >5**	
	Usages	Contextes	Usages	Contextes
GENERAL	8	6,6%	3	5,1%
THEMATIQUE	36	7,4%	1	1,9%
Total	44	14,0%	4	7,0%

Tableau 6. Omissions

Plus gênantes sont les omissions d'usages thématiques, puisqu'elles aboutissent automatiquement à un échec des requêtes. Toutefois, nous remarquons qu'elles sont peu nombreuses.

Si au total 36 usages thématiques sont omis (Tableau 6), 19 ont une occurrence unique dans l'échantillon de test, et un seul a une fréquence inférieure ou égale à 5. Il s'agit d'un usage du mot *lancement*, « lancement d'un programme », qui apparaît 19 fois dans l'échantillon de 100 contextes aléatoires, donc avec une fréquence importante. L'examen détaillé des contextes montre que la thématique pour cet usage est diffuse (bien qu'il ne s'agisse pas d'un usage « général » au sens précédent). Le mot *programme* lui même est rarement présent dans le contexte, et il s'agit généralement d'explications demander de lancer telle ou telle commande, ou application, dans des thématiques extrêmement variés. Il n'est donc pas étonnant que l'algorithme n'ait pas isolé cet usage, pourtant important dans la langue.

Mot-Cible	Racine	Voisins les plus fréquents	Freq (%)[12]
BARRAGE	EAU	construction ouvrage rivière projet retenue crue	67-85
	ROUTIER	véhicule camion membre conducteur policier groupement	3-16
	FRONTIERE	Algérie militaire efficacité armée Suisse poste	5-19
	MATCH	vainqueur victoire rencontre qualification tir football	1-10
DETENTION	PROVISOIRE	juge liberté loi procédure prison instruction	78-93
	DETENU	police centre autorité arrestation torture arbitraire	4-18
	ARME	autorisation acquisition feu munition vente commerce	0-8
	ANIMAL	transport compagnie sauvage certificat annexe directive	0-6
FORMATION	PROFESSIONNEL	centre entreprise organisme stage service programme	96-100
LANCEMENT	SATELLITE	Ariane programme spatial lanceur orbite fusée	94-100
	PRODUIT	public entreprise événement convention presse affaire	0-6
ORGANE	DON	transplantation greffe donneur prélèvement tissu vie	30-51
	DELIBERANT	public établissement président demande attribution communauté	8-24
	REGLEMENT	pays appel différend OMC réunion autorité	12-30
	TECHNIQUE	scientifique convention économique conférence subsidiaire programme	0-8
	CONSULTATIF	matière civil tête supervision mémorandum PAB	1-12
	MALADIE	cœur traitement spécimen preuve sang intervention	0-4
	REPRESENTANT	délégué suprême concertation département personnel agent	0-9
	PARTI	presse chef journal Genève Allemagne rédacteur	0-9
PASSAGE	EURO	public travail entreprise système national monnaie	41-63
	AN_2000	programme autorité installation réseau solution matériel	2-13
	NIVEAU	porte chemin ouverture salle route entrée	0-9
	LIBRE	cour prestation police assurance caisse prévoyance	3-16
	CHEVAL	main énergie équilibre trot dos foulée	0-4
	PARAMETRE	mode appel variable argument langage expression	2-14
	GALERIE	ville boutique bois panorama époque verrière	0-8
	TERRE	durée mouvement soleil Vénus Mercure nœud	4-18
	MORT	rite Dieu naissance Christ vivant Jésus	0-4
RESTAURATION	HOTELLERIE	formation durée centre professionnel entreprise alternance	23-43
	CONSERVATION	sauvegarde atelier monument technique historique oeuvre	34-55
	HEBERGEMENT	activité hôtel région loisir culture contact	0-8
	RAPIDE	restaurant vente établissement repas marche traiteur	1-10
	FICHIER	système information donnée client espace bande	7-23
	PIERRE	bâtiment chantier terre polychromie taille sec	0-4
	MEUBLE	bois table mobilier décoration fabrication antiquité	1-10
SOLUTION	GESTION	entreprise service logiciel client information système	75-91
	JEU	monde gratuit astuce joueur gain francophone	0-4
	INJECTABLE	perfusion glucose HOP commercialisation arrêt Fandre	7-23
STATION	SKI	hiver piste montagne sport village location	75-91
	METEO	température Oregon scientific WS professionnel capteur	0-6
	SPATIAL	international MIR système programme ISS projet	4-18
	TRAVAIL	réseau traitement donnée carte Sun environnement	1-10
	RADIO	navire région réception installation antenne communication	0-6
	PRIMAGAZ	Paris aire Esso province Marseille Dyneff	0-4
	EAU	épuration source mer plage Yves rivière	0-4
	LIGNE	métro quai terminus voyageur correspondance atelier	0-4
VOL	AVION	billet pilote club sec départ voyage	51-72
	LIBRE	école parapente loisir montagne formation Paris	23-43
	VOILE	centre photo vent pilotage forum compétition	0-4
	VOLÉ	service recherche numéro base donnée véhicule	2-13

Tableau 7. Principaux usages pour les mots-test

En résumé, on peut affirmer que le comportement d'*HyperLex* en termes de détection des usages thématiques est tout à fait satisfaisant du point de vue quantitatif : on peut affirmer que la quasi-totalité des usages de fréquence supérieure à 5% est bien détectée.

Du point de vue qualitatif, le découpage proposé est adéquat dans la plupart des cas. Certains découpages émergent, qui ne seraient probablement pas proposés par un lexicographe, et peuvent donc surprendre au prime abord. Ainsi, deux usages sont distingués (entre autres) pour le mot *détention,* repérés par les hubs racines *DETENU* et *PROVISOIRE.* Dans les deux cas, il s'agit de détention de personnes en prison, en opposition aux autres usages détectés (détention d'armes,

[12] Intervalle de confiance à 95% calculé par la loi binomiale.

d'animaux). Cependant, en regardant plus précisément les pages concernées, on s'aperçoit que le sous-corpus contient deux thématiques très disjointes : l'une (hub *DETENU*) concerne les aspects humains de la détention (conditions de détention, torture, visites, etc.), l'autre (hub *PROVISOIRE*) les aspects juridiques (détention provisoire, lois, etc.). Dans une perspective de recherche d'information, il n'est pas illogique de les distinguer, même si on souhaite peut-être un regroupement hiérarchique de certains usages. Le taux de liens inter-composantes, peut-être un bon indice ; en tous cas, il s'agit là d'une piste de recherches ultérieures.

A l'inverse, l'algorithme fusionne les stations de radio à usage public (FM, etc.) et les stations radio de navires, les champs lexicaux étant assez proches (*radio, communication, bande, MHz,* etc.). On voudrait pourtant très certainement différencier ces usages, qui correspondront vraisemblablement à des requêtes différentes. L'algorithme pourrait être amélioré sur plusieurs points, notamment par la prise en compte des distances entre cooccurrents. Ainsi, l'expression « station de radio » n'est utilisée que pour les stations de radio à usage public, alors que pour l'usage maritime, on trouve l'expression « station de navire », et le mot *radio* lui même est généralement plus éloigné dans le contexte, entrant dans d'autres expressions (*opérateur radio, équipement radio,* etc.).

Enfin, dans quelques cas, c'est la description par le hub racine et ses voisins qui n'est pas optimale. Les raisons en sont variées. Par exemple, l'un des usages de *station* est repéré par le hub *PRIMAGAZ,* une marque de gaz G.P.L. La raison en est que de nombreuses listes de stations services délivrant du gaz G.P.L. sont en ligne sur Internet, l'approvisionnement étant manifestement un sujet de préoccupation pour les possesseurs de véhicules utilisant ce type de carburant. G.P.L. serait un hub racine plus approprié, mais si les pages contiennent généralement une première ligne d'en-tête comportant ce mot, les paragraphes contenant le mot *station* ne contiennent que des marques (*Primagaz, Shell, Esso,* etc.) et des adresses. Une analyse plus globale des pages serait peut-être intéressant, au moins pour la labellisation des usages.

2. Etiquetage global

Nous avons appliqué le désambiguïsateur décrit dans la section Désambiguïsation sur les 10 sous-corpus correspondant aux 10 mots-test. Lorsque plusieurs contextes contenaient le mot-test visé sur une même page Web, l'usage le plus fiable (mesuré à l'aide du coefficient de fiabilité ρ) a été appliqué à tous les contextes de la page. Ce coefficient peut aussi servir à régler le rappel de l'étiquetage. Nous avons choisi la valeur $\rho \geq 0.5$, qui correspond à une différence de 1 entre les deux meilleurs scores et permet un rappel de 82%, ce qui semble plus que suffisant pour l'application concernée.

Nous avons tiré au hasard 100 contextes pour chaque mot-cible (1000 au total) parmi ceux ayant $\rho \geq 0.5$, et nous avons vérifié manuellement l'adéquation de l'usage proposé par l'algorithme. Nous avons calculé pour chaque sous-corpus la précision de l'étiquetage, ainsi que la ligne de base (*baseline*) obtenue avec l'étiquetage par l'usage le plus fréquent (Tableau 8). On voit que globalement la précision est de 97%[13], ce qui est excellent, mais il faut toutefois tenir compte de la ligne de base, qui est de 73%. Nous proposons une mesure de réduction d'erreur *RE* qui permet de juger exactement du travail accompli par l'algorithme :

$$RE = \frac{precision - baseline}{1 - baseline}$$

HyperLex réduit de 90,4%[14] l'erreur que l'on obtiendrait pour l'étiquetage trivial avec l'usage le plus fréquent (nous n'avons pas inclus dans ce calcul le mot *formation,* dont le sous-corpus ne fait apparaître qu'un usage, et ne pose donc aucune difficulté). La mesure *RE* fait bien apparaître les mots

[13] Intervalle de confiance à 95% calculé par la loi binomiale : $IC_{95\%}$ = 96-98%.
[14] $IC_{95\%}$ = 86-94%

qui sont plus difficiles que d'autres : *organe* et *passage,* et dans un certaine mesure *solution.* A l'exception de ces trois mots, l'algorithme produit un étiquetage sans erreur.

Mot-test	Precision	Baseline	Reduc. Err.
BARRAGE	1,00	0,77	100,0%
DETENTION	1,00	0,87	100,0%
FORMATION	1,00	1,00	n/a
LANCEMENT	1,00	0,99	100,0%
ORGANE	0,88	0,40	80,0%
PASSAGE	0,88	0,52	75,0%
RESTAURATION	1,00	0,44	100,0%
SOLUTION	0,98	0,84	87,5%
STATION	1,00	0,84	100,0%
VOL	1,00	0,62	100,0%
Total	**0,97**	**0,73**	**90,4%**

Tableau 8. Précision de l'étiquetage

L'étiquetage manuel de référence nous a également permis d'estimer les fréquences des différents usages dans le corpus, que nous avons reporté dans la colonne *Freq* du Tableau 7.

3. Pages les plus pertinentes

Comme nous l'avons mentionné au début de cette section, la mesure classique de précision reflète surtout le comportement de l'algorithme sur l'usage majoritaire. Nous avons donc réalisé une évaluation de la précision sur les 25 meilleurs contextes au regard du coefficient de fiabilité ρ pour *chacun* des usages des mots-test. Cette mesure est bien plus sévère que la précédente, puisqu'elle donne un poids égal à chacun des usages, même les plus rares. Elle est cependant assez réaliste puisqu'elle correspond au comportement d'un moteur de recherche qui catégoriserait les résultats avant présentation à l'utilisateur. Le choix de 25 pages semble suffisant : une étude de Silverstein, Henzinger, Marais, & Moricz (1999) portant sur 150 millions de requêtes soumises à *Altavista* montre que 85.2% d'entre elles sont suivies du seul examen du premier écran de 10 résultats, avec un examen de 1.39 écran en moyenne.

Nous avons donc vérifié manuellement les 25 meilleurs contextes pour chacun des 50 usages du Tableau 7, soit 1245 contextes au total[15].

Mot-test	Contextes	Precision
BARRAGE	100	1,00
DETENTION	100	1,00
FORMATION	25	1,00
LANCEMENT	50	1,00
ORGANE	195	0,86
PASSAGE	225	0,92
RESTAURATION	175	0,94
SOLUTION	75	1,00
STATION	200	1,00
VOL	100	1,00
Total	**1245**	**0,96**

La précision globale est de 95,5%[16]. A nouveau quelques erreurs se produisent sur trois des mots (*organe* et *passage* à nouveau, et *restauration*). A part ces 3 mots, tous les contextes retournés relèvent bien de l'usage adéquat, ce qui une performance appréciable, puisque bon nombre des usages retournés ont une fréquence estimée inférieure à 5%.

[15] Un des usages du mot *organe* ne contenait que 20 contextes.

[16] $IC_{95\%}$ = 94.2 - 96.6%.

8. Conclusion

Nous avons proposé un algorithme efficace de désambiguïsation du sens des mots dans un contexte de recherche d'information. Cet algorithme, *HyperLex,* exploite la structure particulière des graphes de cooccurrences, dont nous avons montré qu'elle est du type des « petits mondes » qui font depuis quelques années l'objet de recherches intensives. Comme dans des méthodes précédemment proposées (vecteurs de mots) l'algorithme extrait automatiquement la liste des usages des mots du corpus (ici le Web), ce qui évite les difficultés liées à l'utilisation d'un dictionnaire préétabli. Toutefois, contrairement aux méthodes précédentes, *HyperLex* permet de détecter des usages de fréquence très faible (de l'ordre de 1%). Une évaluation conduite sur 10 mots-test très polysémiques montre que la grande majorité des usages thématiquement pertinents est bien détectée, tandis que l'étiquetage des mots en contexte fait preuve d'une remarquable précision, permettant notamment une catégorisation des résultats des requêtes de très bonne qualité. Des améliorations sont bien entendu possibles, mais cette étude semble de nature à remettre en cause l'idée reçue selon laquelle les techniques de désambiguïsation sont inutiles voire néfastes en RI. La qualité des résultats obtenus semble constituer une avancée importante en désambiguïsation lexicale, au-delà de la seule application en RI.

Enfin, *HyperLex* est associé à une technique de visualisation et de navigation qui permet à l'utilisateur de naviguer dans le lexique et les thématiques du corpus. Son utilisation par le grand public reste à expérimenter, mais l'outil semble d'ores et déjà utile pour les terminologues et lexicographes et autres utilisateurs spécialisés.

Références

Ahlswede T. E. 1993. Sense Disambiguation Strategies for Humans and Machines. In: Proceedings of the 9th Annual Conference on the New Oxford English Dictionary, Oxford (England), pp. 75-88.

Ahlswede T. E. 1995. Word Sense Disambiguation by Human Informants. In: Proceedings of the Sixth Midwest Artificial Intelligence and Cognitive Society Conference, Carbondale (Illinois), pp. 73-78.

Ahlswede T. E., Lorand D. 1993. The Ambiguity Questionnaire: A Study of Lexical Disambiguation by Human Informants. In: Proceedings of the Fifth Midwest Artificial Intelligence and Cognitive Society Conference, Chesterton (Indiana), pp. 21-25.

Albert, R., Barabási. A.-L. 2002. Statistical mechanics of complex networks. Review of Modern Physics 74:47-97.

Amsler R. A., White J. S. 1979. Development of a computational methodology for deriving natural language semantic structures via analysis of machine-readable dictionaries. Final report on NSF project MCS77-01315. University of Texas at Austin, Austin (Texas).

Barabási, A.-L. , Albert, R. 1999. Emergence of scaling in random networks. Science 286:509–512.

Brandenburg, F. J. 1988. Nice Drawing of Graphs are Computationally Hard. In: Gorney, P., Tauber, M. J. (eds.), Visualization in Human-Computer Interaction, Lecture Notes in Computer Science 439, pp. 1-15. Springer-Verlag, Berlin.

Bruce, R., Wiebe, J. 1998. Word sense distinguishability and inter-coder agreement. In: Proceedings of the 3rd Conference on Empirical Methods in Natural Language Processing (EMNLP-98). Association for Computational Linguistics SIGDAT, Granada (Spain), pp. 53-60.

Di Battista, G., Eades, P., Tamassia, R., Tollis, I. G. 1999. Graph Drawing: Algorithms for the Visualization of Graphs. Prentice Hall, New York.

Fellbaum, C., Grabowski, J., Landes, S. 1998. Performance and confidence in a semantic annotation task. In: Fellbaum, C. (ed.) WordNet:An electronic database. Cambridge (Massachusetts). The MIT Press, pp 217-237.

Harris, Z. S. 1954. Distributional Structure. Word 10:146-162.

Hornby, A. S. 1954. A guide to patterns and usage in English. London, OUP.

Hornby, A.S., Gatenby, E.V., Wakefield, H. 1942. Idiomatic and Syntactic English Dictionary. [Photographically reprinted and published as A Learner's Dictionary of Current English by Oxford University Press, 1948; subsequently, in 1952, retitled The Advanced Learner's Dictionary of Current English.] Kaitakusha, Tokyo.

Ide, N. M., Véronis, J. 1998. Introduction to the special issue on word sense disambiguation :the state of the art. Computational Linguistics 24(1):1-40.

Jansen, B. J., Spink, A., Saracevic, T. 2000. Real life, real users, and real needs: A study and analysis of user queries on the web. Information Processing and Management 36(2):207-227.

Jorgensen, J. 1990. The psychological reality of word senses. Journal of Psycholinguistic Research 19:167-190.

Kilgarriff A 1998 SENSEVAL:An Exercise in Evaluating Word Sense Disambiguation Programs. In Proceedings of the Language Resources and Evaluation Conference. Granada (Spain), pp 581-588.

Krovetz, R, Croft, W. B. 1992. Lexical Ambiguity and Information Retrieval. ACM Transactions on Information Systems 10(2):115-141.

Kruskal, J. B. 1956. On the shortest spanning subtree of a graph and the traveling salesman problem. In: Proceedings of the American Mathematical Society, volume 7, pp. 48-50.

Meillet, A. 1926. Linguistique historique et linguistique générale. Vol. 1. Champion, Paris, 351pp. (2nd edition).

Milgram, S. 1967. The small world problem. Psychology Today 2:60–67.

Munzner, T. 2000. Interactive visualization of large graphs and networks. Ph. D. Dissertation, Stanford University. Stanford (California).

Newman, M. E. J. 2003. The structure and function of complex networks. SIAM Review 45:167-256.

Reymond, D. 2002. Dictionnaires distributionnels et étiquetage lexical de corpus. In: Actes de TALN/RECITAL'2001, Atala, Tours (France), pp. 479-488.

Salton, G., McGill, M. 1983. Introduction to Modern Information Retrieval. McGraw-Hill, New York.

Sanderson, M. 1994. Word sense disambiguation and information retrieval. In: Proceedings of the 17th ACM SIGIR Conference, Dublin (Ireland), pp. 142-151.

Schütze, H. 1998. Automatic word sense discrimination. Computational Linguistics 24(1):97-124.

Schütze, H. Pedersen, J. 1995. Information retrieval based on word senses. In: Proceedings of the 4th Annual Symposium on Document Analysis and Information Retrieval, Las Vegas (Nevada), pp. 161-175.

Silverstein, C., Henzinger, M., Marais, H., Moricz, M. 1999. Analysis of a Very Large AltaVista Query Log. SRC Technical note #1998-14. [On-line at http://gatekeeper.dec.com/pub/DEC/SRC/technical-notes/abstracts/src-tn-1998-014.html]

Spink, A., Wolfram, D., Jansen, B. J., Saracevic, T. 2001. Searching of the web: the public and their queries. Journal of the American Society of Information Science and Technology 52(3):226 - 234.

Thorndike, E. L., Lorge, I. 1938. Semantic counts of English Words, Columbia University Press, New York.

Véronis, J. 1998. A study of polysemy judgements and inter-annotator agreement, In: Programme and advanced papers of the Senseval workshop, Herstmonceux Castle (England), pp. 2-4. [Online at http:/www.up.univ-mrs.fr/veronis/pdf/1998senseval.pdf].

Véronis, J. 2001. Sense tagging: does it make sense? Paper presented at the Corpus Linguistics'2001 Conference, Lancaster, U.K. [Online at http://www/up.univ-mrs.fr/veronis/pdf/2001-lancaster-sense.pdf]

Voorhees E. M. 1993. Using WordNet to disambiguate word sense for text retrieval. In: Proceedings of ACM SIGIR Conference, pp. 171-180.

Wallis P. C. 1993. Information retrieval based on paraphrase. In: Proceedings of the First Pacific Association for Computational Linguistics Conference, PACLING Conference, Vancouver (Canada).

Watts, J.W., Strogatz, S.H. 1998. Nature 393:440-442.

Weiss, S.F. 1973. Learning to disambiguate. Information Storage and Retrieval 9:33-41.

Wittgenstein, L. 1953. Philosophische Untersuchungen [Philosophical Investigations, translated by G.E.M. Anscombe, New York, Macmillan.]