État de l'art : mesures de similarité sémantique locales et algorithmes globaux pour la désambiguïsation lexicale à base de connaissances

Andon Tchechmedjiev
LIG-GETALP
Laboratoire d'Informatique de Grenoble-Groupe d'Étude pour la Traduction Automatique/Traitement Automatisé des Langues et de la Parole
Université de Grenoble
andon.tchechmedjiev@imag.fr

RÉSUMÉ

Dans cet article, nous présentons les principales méthodes non supervisées à base de connaissances pour la désambiguïsation lexicale. Elles sont composées d'une part de mesures de similarité sémantique locales qui donnent une valeur de proximité entre deux sens de mots et, d'autre part, d'algorithmes globaux qui utilisent les mesures de similarité sémantique locales pour trouver les sens appropriés des mots selon le contexte à l'échelle de la phrase ou du texte.

ABSTRACT

State of the art : Local Semantic Similarity Measures and Global Algorithmes for Knowledge-based Word Sense Disambiguation

We present the main methods for unsupervised knowledge-based word sense disambiguation. On the one hand, at the local level, we present semantic similarity measures, which attempt to quantify the semantic proximity between two word senses. On the other hand, at the global level, we present algorithms which use local semantic similarity measures to assign the appropriate senses to words depending on their context, at the scale of a text or of a corpus.

MOTS-CLÉS : désambiguïsation lexicale non-supervisée, mesures de similarité sémantique à base de connaissances, algorithmes globaux de propagation de mesures locales.

KEYWORDS: unsupervised word sense disambiguation, knowledge-based semantic similarity measures, global algorithms for the propagation of local measures.

1 Introduction

Les ambiguïtés font partie intégrante des langues naturelles, mais les humains ont la capacité, dans la plupart des cas et en s'aidant du contexte, à désambiguïser sans trop d'efforts. Cependant, pour le traitement automatique des langues naturelles, cette ambiguïté pose problème, et il est fondamental de trouver des méthodes pour affecter aux mots les sens corrects vis à vis du contexte.

Il existe différentes approches pour résoudre ce problème. Elle se divisent en deux catégories principales : d'une part les approches supervisées, nécessitant des corpus d'entraînement étiquetés manuellement et, d'autre part, des approches non-supervisées (Navigli, 2009).

Actes de la conférence conjointe JEP-TALN-RECITAL 2012, volume 3: RECITAL, pages 295–308,
Grenoble, 4 au 8 juin 2012.

Le problème avec les algorithmes supervisés est le fait qu'obtenir de grandes quantités de texte annoté en sens est très couteux en temps et en argent, et que l'on se heurte au goulot d'acquisition de données (Wagner, 2008). De plus, la qualité de la désambiguisation de ces approches est restreinte par les exemples utilisés pour l'entrainement.

C'est pourquoi les méthodes non supervisées sont intéressantes. Elles n'utilisent pas de corpus annotés. Il existe là aussi des distinctions : d'une part les approches non supervisées classiques (clustering) qui exploitent les données non annotées ; et d'autre part les approches à base de savoirs qui utilisent des connaissances issues de ressources lexicales.

Nous nous intéressons ici à ces dernières. Il y a différents aspects à considérer dans le cadre des approches non-supervisées à base de connaissances : d'abord la question essentielle des ressources lexicales qu'il est possible d'utiliser, ensuite la question de comment exploiter la, ou les ressources lexicales pour désambiguïser.

Ce dernier aspect se présente sous deux dimensions : la dimension locale où l'on cherche à déterminer la proximité entre les sens possibles des différents mots et, la dimension globale où l'on cherche à affecter les bons sens aux mots à l'échelle d'un texte. Il existe à la fois des méthodes qui *propagent* les mesures locales en les utilisant pour évaluer les combinaisons de sens, mais aussi des méthodes purement globales qui exploitent directement la structure linguistique de l'ensemble du texte sans s'intéresser aux sens individuellement.

Nous présenterons, les principales ressources lexicales (Section 2), les principales mesures de similarité sémantique (Section 3) puis une description de quelques algorithmes globaux qui exploitent ces mesures (Section 4). Nous terminerons par des considérations sur l'évaluation et la comparaison de ces algorithmes (Section 5).

Pour un état de l'art complet et plus détaillé, le lecteur se référera à (Ide et Veronis, 1998) et plus récemment (Navigli, 2009).

2 Ressources lexicales

Une caractéristique des approches à base de connaissances est qu'elles utilisent des ressources lexicales. Un premier type de ressource qui peut être exploitée est l'inventaire de sens, c'est-à-dire une ressource qui, à chaque mot, lie une liste de sens possibles comme par exemple, un dictionnaire (par exemple (Collins, 1998)). D'autre part, des ressources telles que les thésaurus (par exemple (Roget, 1989)) peuvent être utiles pour établir des liens entre les sens des différents mots.

Par ailleurs, des ressources lexicales telles que WordNet (Miller, 1995) sont structurées et jouent le rôle d'inventaires de sens et de dictionnaires, mais donnent également accès à une hiérarchie de sens (en quelque sorte un thésaurus structuré).

La majorité des mesures de similarité que nous allons présenter se basent sur Wordnet [1].

WordNet est structuré autour de la notion de synsets, c'est-à-dire en quelque sorte un ensemble de *synonymes* qui forment un concept. Un synset représente un sens de mot. Les synsets sont reliés entre eux par des relations, soit lexicales (antonymie par exemple) ou taxonomiques (hyperonymie, méronymie, etc).

1. Il est possible de les utiliser sur d'autres ressources également.

3 Mesures de similarité Sémantique à base de connaissances

Parmi les mesures de similarité sémantique on retrouve trois types principaux que nous allons maintenant décrire. Il faut noter que les mesures de de similarité géométriques ne sont pas à base de connaissances et ne seront pas présentées.

3.1 À base de traits

3.1.1 Similarité de Tsversky

Avant d'être abordée en TALN, la notion de similarité sémantique a été traitée dans le domaine de la psychologie cognitive. Un travail souvent cité est (Tversky, 1977) qui propose une nouvelle approche basée sur le recouvrement ou non de traits entre deux objets. Plus précisément, est considéré comme concept ou signification rattachée à un objet toute propriété dudit objet. La similarité entre deux objets est exprimée comme le nombre pondéré de propriétés en commun, auxquelles on retire le nombre pondéré de propriétés spécifiques à chaque objet. Il propose donc un modèle de similarité non symétrique, que l'on appelle «modèle de contraste» (Figure 1).

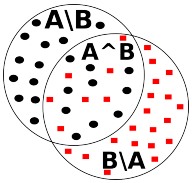

FIGURE 1 – Le modèle de contraste entre deux concepts

Plus formellement, si l'on reprend la notation de (Pirró et Euzenat, 2010) où $\Psi(c)$ est l'ensemble des traits se rapportant à un sens s, alors la similarité de Tsversky peut s'exprimer par : $sim_{tvr}(s_1,s_2) = \theta F(\Psi(s_1) \cap \Psi(s_2)) - \alpha F(\Psi(s_1) \setminus \Psi(s_2)) - \beta F(\Psi(s_2) \setminus \Psi(s_1))$ où F est une fonction qui associe une pertinence aux traits, et où θ, α et β sont des facteurs qui marquent respectivement l'importance relative de la similarité entre les sens, des dissimilarités entre s_1 et s_2 et des dissimilarités entre s_2 et s_1, et où $\setminus$ est l'opérateur de différence ensembliste.

Il est également possible d'exprimer cette mesure en tant que rapport, afin d'avoir une valeur de similarité normalisée (avec $\theta = 1$) :,

$$sim_{tvtr}(c_1,c_2) = \frac{F(\Psi(c_1) \cap \Psi(c_2))}{F(\Psi(c_1) \cap \Psi(c_2)) + \alpha F(\Psi(c_1) \setminus \Psi(c_2)) + \beta F(\Psi(c_2) \setminus \Psi(c_1))}$$

Comme récapitulé dans (Pirró et Euzenat, 2010), différentes valeurs de α et de β mènent à différents types de similarité. Si $\alpha = \beta = 0$, on ne s'intéresse qu'aux points communs entre les deux sens. Si $\alpha > \beta$ ou $\alpha < \beta$ alors on s'intéresse assymétriquement à la similarité de s_1 avec s_2 ou vice versa. Si $\alpha = \beta \neq 0$ on s'intéresse à la similarité mutuelle entre s_1 et s_2. Quand $\alpha = \beta = 1$ la mesure de Tversky est équivalente à la similarité de Tanimoto (Rogers et Tanimoto, 1960). Dans le cas où $\alpha = \beta = 0.5$ alors elle est équivalente au coefficient de Dice (Dice, 1945).

3.1.2 Similarité de Lesk

(Lesk, 1986) a proposé un algorithme de désambiguïsation lexicale très simple, qui considère la similarité entre deux sens comme le nombre de mots en commun dans leurs définitions. Dans la version originale, on ne prend pas en compte l'ordre des mots dans les définitions (sac de mots). Dans ce cadre là, il apparaît que cette méthode puisse être ramenée à un cas particulier de la similarité de Tsversky (en tant que rapport ou non), en considérant que les concepts sont des sens de mots, que les traits sont des mots de la définition des sens, avec $\alpha = \beta = 0$, et avec $\Psi(s) = D(d)$ qui retournant un ensemble contenant les mots de la définition d'un sens de mot s. Quant à la fonction F on la choisit comme la fonction cardinalité d'ensemble. On obtient ainsi : $sim_{lesk}(s_1, s_2) = \mid D(s_1) \cap D(s_2) \mid$

L'avantage de cette mesure de similarité est qu'elle est extrêmement simple à calculer, et ne requiert qu'un dictionnaire. Dans le contexte de l'algorithme de Lesk original, la similarité était calculée de manière exhaustive entre tous les sens de tous les mots du contexte, il existe une variante (Navigli, 2009) utilisée sur une fenêtre de contexte autour du mot auquel appartient le sens. Elle correspond au recouvrement entre la définition du sens et entre un sac de mot contenant tous les mots des définitions des mots du contexte : $Lesk_{var} = \mid contexte(w) \cap D(s_{w_n}) \mid$. Comme le met en avant (Navigli, 2009), un problème important de la mesure de Lesk est qu'elle est très sensible aux mots présents dans la définition, et si certains mots importants manquent dans les définitions utilisées, les résultats obtenus seront de qualité moindre. De plus si les définitions sont trop concises (comme c'est souvent le cas) il est difficile d'obtenir des distinctions de similarité fines.

Cependant, un certain nombre d'améliorations de la mesure de Lesk ont été proposées.

3.1.3 Extensions de la mesure de Lesk

Tout d'abord, (Wilks et Stevenson, 1998) proposent de pondérer chaque mot de la définition par la longueur de celle-ci afin de donner la même importance à toutes les définitions, au lieu de systématiquement privilégier les définitions longues.

Plus récemment (Banerjee et Pedersen, 2002) ont proposé la mesure de "Lesk étendu", qui améliore Lesk de deux façons. La première est l'incorporation des définitions des sens reliés par des relations taxonomiques WordNet dans la définition d'un sens donné. La deuxième est une nouvelle manière de calculer le recouvrement entre les mots des définitions.

Pour calculer le recouvrement entre deux sens, ils proposent de considérer le recouvrement entre les définitions des deux sens mais aussi des définitions issues de différentes relations : hyperonymes (`has-kind`), hyponymes (`kind-of`), meronymes (`part-of`), holonymes (`has-part`), troponymes mais aussi par les relations `attribute`, `similar-to`, `also-see`.

Afin de garantir que la mesure soit symétrique, ils proposent de prendre les combinaisons deux à deux des relations considérées et de ne conserver une paire de relations (R_1, R_2) que si la paire inverse (R_2, R_1) est présente. On obtient ainsi un ensemble $RELPAIRS$. De plus, le recouvrement entre deux définitions A et B se calcule comme la somme des carrés des longueurs de toutes les sous-chaines de mots de A dans B, ce que l'on exprime avec l'opérateur $\Cap$. Nous avons ainsi : $Lesk_{etendu}(s_1, s_2) = \sum_{\forall (R_1,R_2) \in RELPAIRS^2} (\mid D(R_1(s_1)) \Cap D(R_2(s_2)) \mid)$

Le calcul du recouvrement est basé sur le principe relevé par la loi de Zipf (Zipf, 1949), qui met en évidence une relation quadratique entre la longueur d'une phrase et sa fréquence d'occurrence dans un corpus. De ce fait, n mots qui apparaissent ensemble portent plus d'informations que si ils étaient séparés.

3.2 À base de distance taxonomique

Le principe des mesures à base de distance taxonomique est de compter le nombre d'arcs qui séparent deux sens dans une taxinomie.

La Figure 2 (Wu et Palmer, 1994) représente la relation de deux sens quelconques S_1 et S_2 dans une taxinomie par rapport à leur sens commun le plus spécifique S_3 et par rapport à la racine de la taxinomie ; cette figure servira à exprimer de manière homogène les formules des différentes mesures de similarité.

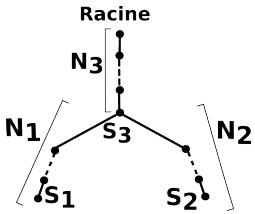

FIGURE 2 – Deux sens et leur sens commun le plus spécifique dans une taxinomie

La mesure de Rada (Rada *et al.*, 1989) est la première à utiliser la distance entre les nœuds correspondant aux deux sens sur les liens d'hyponymie et hyperonymie :

$$Sim_{Rada}(s_1,s_2) = d(s_1,s_2) = N_1 + N_2$$

Les termes se trouvant plus profondément dans la taxonomie étant toujours plus proches que les termes plus généraux, (Wu et Palmer, 1994) proposent de prendre en compte la distance entre l'ancêtre commun le plus spécifique et la racine pour y remédier.

$$Sim_{WuP} = \frac{2 \cdot N_3}{N_1 + N_2 + 2 \cdot N_3}$$

(Leacock et Chodorow, 1998) se basent également sur la mesure de Rada, mais au lieu de normaliser par la profondeur relative de la taxinomie par rapport aux sens, ils choisissent une normalisation par rapport à la profondeur totale de la taxinomie D et normalisent avec un logarithme :

$$Sim_{LCH} = -log(\frac{N_1 + N_2}{2 \cdot D})$$

(Hirst et St-Onge, 1998) adaptent le concept de chaines lexicales développées par (Morris et Hirst, 1991) comme mesure de similarité sémantique en utilisant la structure de WordNet. Cette mesure se base sur l'idée de (Halliday et Hasan, 1976) que dans un texte, des mots ont une forte probabilité de référer à des mots antérieurs ou à d'autres concepts reliés, et que l'enchainement de ces mots forment des chaines cohésives. Par exemple, (Navigli, 2009) `Rome->ville->habitant` et `manger->plat->légume->aubergine`, sont des chaines lexicales.

À chaque relation est associée une direction horizontale, ascendante ou descendante, qui marque respectivement une relation forte, très forte et moyennement forte (par exemple l'hyperonymie

est une relation ascendante, l'holonymie une relation descendante, et l'antonymie une relation horizontale). Les changements de direction constituent un élément de dissimilarité et la proximité dans la taxinomie un élément de similarité. Un changement de direction est défini comme le passage d'un élément A de la taxinomie à un élément B par une relation d'un autre type que celle qui a permis d'arriver sur A. Notons que plus la distance entre les sens est grande, plus il y aura de changements de direction potentiels.

Même si l'algorithme proposé est un algorithme global, il est possible d'utiliser la fonction d'évaluation des chaines lexicales en tant que mesure de similarité.

Soient C et k deux constantes et soit la fonction $virages(s_1, s_2)$ qui retourne le nombre de changements de direction entre les sens s_1 et s_2, alors la mesure de similarité s'exprime :

$Sim_{Hso} = C - (N_1 + N_2) - k \cdot virages(s_1, s_2)$

Il existe également d'autres mesures exploitant la structure taxonomique, mais en pondérant les arcs avec des valeurs de contenu informationnel, notion que nous allons maintenant définir.

3.3 À base de contenu informationnel

L'approche de (Resnik, 1995) est basée sur la détermination du contenu informationnel du concept commun le plus spécifique à deux sens. Dans la figure 2, le concept commun le plus spécifique à S_1 et S_2 est S_3 (notée $lso(S_1, S_2) = S_3$ pour *lowest superordinate*). Quant à la quantité d'information, elle est calculée à partir de la probabilité $p(S_3)$: $IC(s) = -log(P(S_3))$

Les probabilités d'occurrence de chaque concept de la taxinomie sont calculées à partir d'un corpus non-annoté par estimation du maximum de vraisemblance. Ainsi la mesure similarité de Resnik s'exprime :

$$Sim_{Res} = IC(lso(S_1, S_2)) = IC(S_3)$$

(Jiang et Conrath, 1997) partent du constat qu'utiliser seulement l'ancêtre commun n'offre pas une granularité assez fine et proposent de prendre en compte la quantité d'information portée par les deux sens. Cette mesure s'exprime :

$$Sim_{JCN} = IC(S_1) + IC(S_2) + 2 \cdot IC(lso(S_1, S_2))$$

(Lin 1998) propose également un mesure de similarité très proche, qui revient essentiellement à une reformulation sous forme de rapport de la formule de Jiang and Conrath :

$$Sim_{Lin} = \frac{2IC(lso(S_1, S_2))}{IC(S_1) + IC(S_2)}$$

Plus récemment, (Seco *et al.*, 2004) ont argumenté qu'il est possible d'extraire directement de WordNet les valeurs de contenu informationnel sans avoir à passer par un corpus. La taxinomie de WordNet étant structurée à partir de principes psycholinguistiques, on peut faire l'hypothèse que cette structure (liens d'hyperonymie et d'hyponymie) est représentative du contenu informationnel ; c'est-à-dire que les concepts qui ont beaucoup d'hyponymes portent une quantité d'information moins importante que les concepts *feuilles*. Si l'on note $hypo(s)$ une fonction qui retourne le nombre d'hyponymes d'un concept s et max_{wn} le nombre total

de concepts dans la taxinomie, alors on peut exprimer le contenu informationnel intrinsèque : $iIC(s) = 1 - \frac{log(hypo(s)+1)}{log(max_{wn})}$

On peut substituer iIC à IC dans toutes les mesures précédentes pour supprimer le besoin d'apprentissage non supervisé. Cependant, l'iIC est limitée car, elle n'exploite qu'un seul type de relations, alors que d'autres pourraient être intéressantes. C'est pourquoi (Pirró et Euzenat, 2010) proposent une mesure d'iIC étendue qui va prendre en compte les autres relations présentes (meronymie par exemple). Ils l'expriment comme : $eIC(s) = \zeta iIC(s) + \eta EIC(s)$

où ζ et η sont des constantes et où $EIC(s)$ est la somme des iIC moyennes des concepts reliés à s par les autres relations. Si l'on note $Rels(s)$ l'ensemble des relations possibles pour s, et si pour une relation $s_r \in Rels(s)$, $s_r(s)$ est l'ensemble des concepts reliés à s par s_r, alors on obtient : $EIC(s) = \sum_{s_r \in Rels(s)} \frac{\sum_{c \in s_r(s)} iIC(c)}{|s_r(s)|}$

Il est également possible de substituer eIC à IC.

Plus récemment ont commencé à apparaître des mesures hybrides qui, soit essayent de combiner différents types de mesures de similarité, soit essayent d'exploiter la structure de plusieurs ontologies (*cross-ontology similarity*).

3.4 Mesures hybrides

Ici, nous nous focalisons sur l'aspect combinaisons de mesures plutôt que sur les approches à ontologies croisées.

(Li et al 2003) proposent une mesure qui combine à la fois la distance taxonomique ($l = N_1 + N_2$), la profondeur du concept commun le plus spécifique dans la taxinomie ($h = N_3$) ainsi que la densité sémantique locale($d = IC(lso(s_1, s_2))$) , cette dernière étant exprimée en terme de contenu informationnel. Leur mesure est exprimée par : $Sim_{Li}(s_1, s_2) = f(f_1(l), f_2(h), f_3(d))$ où f_1, f_2 and f_3 sont les fonctions de transfert non-linéaires respectives pour chaque type d'information.

Le but des fonctions de transfert est de normaliser dans l'intervalle $[0; 1]$ les mesures pour qu'elles puissent être combinées. $f_1(l) = e^{-\alpha l}$ où α est une constante. $f_2(h) = (e^{\beta h} - e^{-\beta h}) \div (e^{\beta h} + e^{-\beta h})$ où $\beta > 0$ est un facteur de lissage. $f_3(d) = (e^{\lambda d} - e^{-\lambda d}) \div (e^{\lambda d} + e^{-\lambda d})$ avec $\lambda > 0$. Quant à la fonction f, elle constitue n'importe quelle combinaison de ces trois mesures, et est à choisir selon les applications et la nature des sources d'information disponibles.

FaITH, une autre mesure locale qui combine des aspects de différents types, est proposée par (Pirró et Euzenat, 2010) sous la forme de l'extension des mesures à base de contenu informationnel en reprenant le modèle de contraste de Tsversky :

$$Sim_{FaITH} = \frac{IC(lso(s_1, s_2))}{IC(s_1) + IC(s_2) - IC(lso(s_1, s_2))}$$

Ici la fonction F est remplacée par IC, les traits communs aux concepts sont représentés par le contenu informationnel du concept commun le plus spécifique, et les traits spécifiques à un concept sont représentés par la différence entre le contenu informationnel de ce concept auquel on soustrait le contenu informationnel du concept commun le plus spécifique.

4 Algorithmes globaux de désambiguïsation lexicale

Maintenant que nous avons passés en revue les principales de mesures de similarité sémantique, nous allons présenter différents algorithmes qui les utilisent comme heuristiques pour évaluer des combinaisons de sens.

4.1 Approche exhaustive

L'approche originellement adoptée par (Lesk, 1986) pour désambiguïser un texte en entier, est d'évaluer toutes les combinaisons possibles de sens et de choisir la combinaison qui maximise le score du texte – exprimé comme la somme des scores des sens choisis par rapport aux autres mots du texte.

En d'autres termes, si le sens sélectionné d'un mot w dans une combinaison C est S_w et un texte T une liste ordonnée de mots, alors le score de la combinaison est $score(C) = \sum_{w_i \in T} \sum_{w_j \in T} sim(S_{w_i}, S_{w_j})$ et il y a en tout $\prod_{w \in T} N_w$ combinaisons à évaluer, avec N_w le nombre de sens de w (Gelbukh *et al.*, 2005), c'est-à-dire un nombre exponentiel de combinaisons.

Par exemple pour une phrase de 10 mots avec 10 sens en moyenne par mot il y aurait $10^{10^2} = 10^{100}$ combinaisons.

Pour diminuer le temps de calcul on peut utiliser une fenêtre autour du mot afin de réduire le temps d'évaluation d'une combinaison au prix d'une perte de cohérence globale de la désambiguïsation.

Une autre approche est d'utiliser des meta-heuristiques d'optimisation combinatoire pour obtenir des solutions de qualité convenable d'une manière qui soit traitable calculatoirement.

4.2 Recuit simulé

Le recuit simulé est une méthode d'optimisation stochastique classique, et fût appliqué à la désambiguïsation lexicale par (Cowie *et al.*, 1992).

Le principe est de faire des changements aléatoires dans la configuration [2] itérativement de l'espace de recherche puis d'évaluer si le changement est bénéfique. Dans le cas échéant, il est conservé, sinon, il y a une probabilité de le conserver quand même.

Cette évaluation se fait en utilisant une métrique heuristique, et dans le cas de la désambiguïsation, on utilise les mesures de similarité pour jouer ce rôle. Le score d'une configuration se calcule de la même manière que pour l'évaluation d'une combinaison dans le cas de l'approche exhaustive.

Quant à la probabilité de conserver une configuration inférieure, elle se calcule par rapport à la différence des scores entre la configuration modifiée(C') et la configuration avant modification (C) avec $\Delta s = score(C') - score(C)$ et un paramètre de température T : $P(conservation) = e^{\frac{-\Delta s}{T}}$.

Dans le cas d'une descente par gradients où l'on ne garde que les meilleures configurations, on est confronté à un problème de convergence sur des maxima locaux. L'objectif de l'acceptation possible des configurations inférieures pour le recuit simulé est de leur échapper. Cependant cela pourrait mener à une non convergence du système, c'est pourquoi la diminution géométrique de la température T permet progressivement de se ramener à une descente de gradient, dont la convergence est garantie.

2. Une configuration est représentée par un vecteur d'entiers correspondant aux numéros de sens sélectionnés pour chaque mot dans l'ordre d'apparition des mots dans le texte.

4.3 Algorithme génétique

Les algorithmes génétiques sont inspirés de l'évolution génétique des espèces et sont utilisés pour trouver des solutions à des problèmes d'optimisation combinatoire.

(Gelbukh *et al.*, 2003) l'ont appliqué à la désambiguïsation lexicale. La représentation de la configuration utilisée est la même que pour le recuit simulé, cependant, on considère une population de λ configurations (chromosomes). Chaque indice du vecteur d'une configuration est considéré comme un allèle, et les allèles possibles pour un indice sont les différents sens du mot en question.

Le déroulement de l'algorithme est inspiré des cycles reproductifs et de sélection naturelle des espèces. La *qualité* d'un individu est estimée avec une fonction de score heuristique, ici la même que pour le recuit simulé.

À chaque cycle, les scores de tous les individus sont calculés, et un nombre pair d'individus sont sélectionnés de manière probabiliste pour être croisés : $\forall \lambda_i \in \lambda,\ p(crois_{\lambda_i}) = Cr * \left(\frac{score(\lambda_i)}{score_{max}}\right)$, où $score_{max}$ est le meilleur score dans la population et où Cr est un rapport de sélection.

Le croisement s'effectue par une permutation autour d'un ou plusieurs points de pivots choisis au hasard dans la configuration, habituellement un ou deux. Les individus non retenus pour le croisement sont dupliqués. On obtient ainsi une nouvelle population (qui remplace l'ancienne). Sur chaque individu, on applique avec une probabilité $p(M)$, Mn changements aléatoires uniformes. Le score de la nouvelle population est calculé après la phase de mutation, puis un nouveau cycle commence.

Parmi les stratégies de convergence, on trouve un nombre fixe de cycles où encore une stabilisation de la distribution des scores de la population pendant plusieurs cycles successifs.

4.4 Chaines lexicales

Comme décrit dans la Section 3.2 (Hirst et St-Onge, 1998) est un algorithme global qui se base sur la construction de chaines lexicales afin d'évaluer les combinaisons en intégrant des connaissances linguistiques pour réduire l'espace de recherche.

Ils placent tout d'abord des restrictions sur les enchainements de types de liens possibles : il est impossible d'avoir plus d'un changement de direction ; un lien ascendant est terminal, sauf si il est suivi d'un lien horizontal faisant le lien avec un lien descendant.

La chaine lexicale globale est construite dans l'ordre des mots du le texte. Lorsqu'un mot est inséré (présent dans le texte, ou transitivement par une relation), un certain nombre de ses *synsets* lui sont reliés : si c'est le premier mot de la chaine ou si le mot provient d'une relation très forte alors on garde tous les *synsets* ; quand il provient d'un lien fort, alors on inclut seulement les *synsets* qui lui sont reliés par des liens forts ; et quand le mot provient d'une relation moyennement forte alors on ne considère que les *synsets* avec le meilleur score (avec leur mesure locale).

Tous les *synsets* qui ne participent pas à une relation selon les critères précédents sont supprimés. De plus, au fur et à mesure que des mots sont ajoutés à la chaine, et si elle devient illégale, tous ses synsets sont supprimés.

Par ailleurs quand on insère un mot, on cherche d'abord parmi les relations par ordre décroissant de force et dans un contexte de plus en plus petit (respectivement, toutes les phrases, sept

phrases, trois phrases) si une chaine existe déjà, auquel cas le mot y est ajouté ; dans le cas contraire une nouvelle chaine est créée.

Le problème de cette méthode est qu'elle est peu précise à cause de sa nature gloutonne (Navigli, 2009) et différentes améliorations ont été proposées. On peut citer (Barzilay et Elhadad, 1997) qui gardent toutes les interprétations possibles, ce qui augmente la précision au détriment des performances. (Silber et McCoy, 2000) proposent un algorithme de construction de chaines lexicales linéaire, qui permet de résoudre le problème de performance tout en conservant la qualité accrue.

4.5 Algorithme à base d'exploration pseudo aléatoire de graphes

D'autres algorithmes sont ceux qui se basent sur le principe d'une marche aléatoire dans un graphe, ce qui inclut à la fois des algorithmes de type *PageRank*, mais aussi des méta-heuristiques à colonies de fourmis.

4.5.1 Algorithme à base de PageRank

(Mihalcea et al. 2004) ont appliqué l'algorithme de *PageRank* (Brin et Page, 1998) pour la désambiguïsation lexicale. Le principe est d'assigner des poids aux arcs d'un graphe récursivement en exploitant les informations globalement disponibles.

(Mihalcea et al. 2004) utilisent WordNet et ses relations pour construire un graphe dirigé [3] ou non représentant les différentes combinaisons de sens à partir du texte à désambiguïser. Pour chaque mot, l'ensemble des *synsets* qui lui sont liés constituent les nœuds du graphe, alors que les arcs sont les relations issues de Wordnet (ou d'une combinaison de relations) entre les *synsets* des mots du texte. À noter que les *synsets* du même mot ne sont jamais reliés entre eux.

Une fois le graphe construit, un poids est associé à chaque arc. Le choix des poids initiaux peut se faire de deux manières : initialisés à 1, ou par une mesure de similarité sémantique.

S'en suit la marche aléatoire du PageRank. Le marcheur choisit l'arc à emprunter de manière aléatoire suivant la distribution des valeurs de PageRank des nœuds reliés au nœud courant. À chaque passage sur un nœud, le score est mis à jour avec : $S(N_i) = (1-d)+d*\sum_{j \in In(N_i)} \frac{S(N_i)}{|Out(N_i)|}$, où d est un facteur de lissage, N_i un nœud du graphe, $In(N_i)$ l'ensemble des nœuds prédécesseurs de N_i et $Out(N_i)$ l'ensemble des nœuds successeurs. Lorsque le système a convergé sur la distribution stationnaire, le sens de chaque mot correspondant au nœud avec le meilleur score PageRank est sélectionné le sens.

4.5.2 Algorithme à colonies de fourmis

Les algorithmes à colonies de fourmis sont des algorithmes multi-agents réactifs qui cherchent à imiter le fonctionnement d'une colonie de fourmis. Cette approche fut initialement proposée par (Dorigo, 1992), et part de la constatation faite par (Deneubourg *et al.*, 1983) que les fourmis, lorsque plusieurs chemins sont possibles pour atteindre de la nourriture, convergent systématiquement vers le chemin le plus court. Lors de leur passage, les fourmis déposent une substance chimique (phéromone) pour alerter les autres fourmis de la présence de nourriture ;

3. Sachant que les relations dans Wordnet son symétriques, on retiens arbitrairement soit la relation, soit son inverse

cette substance s'évapore si elle n'est pas renforcée par le passage d'autres fourmis. C'est cette communication indirecte au travers de l'environnement (stigmergie) qu'émerge une convergence optimale du système. L'utilisation d'un tel algorithme pour la désambiguïsation lexicale a été proposée par (Guinand et Lafourcade, 2010) en utilisant un modèle de similarité à base de vecteurs pour régir le déplacement des fourmis. Plus récemment (Schwab *et al.*, 2011) ont proposé de remplacer ce modèle par l'utilisation d'une approche basée sur la mesure de Lesk étendu.

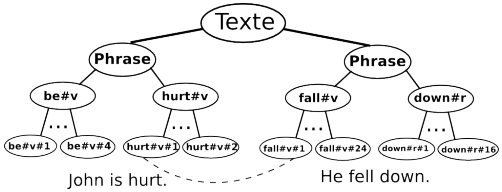

FIGURE 3 – La structure de l'environnement de l'algorithme à colonies de fourmis

Le graphe, contrairement à celui de (Mihalcea *et al.*, 2004), ne lie pas les sens entre eux, mais reprends une structure d'arbre qui suit celle du texte [4].

Nous pouvons voir sur la Figure 3 un exemple de graphe pour une phrase simple. La racine est un nœud correspondant au texte. Les nœuds fils sont des nœuds correspondant aux phrases ; leurs nœuds fils correspondant aux mots les feuilles des mots correspondent aux sens. Ces nœuds produiront les fourmis (fourmilières). Au départ il n'y a aucune connexion entre les nœuds "sens" des différents mots, ce sont les fourmis qui vont créer des "ponts" entre eux. Chaque nœud qui n'est pas une fourmilière possède un vecteur de sens qui lui est attaché (vecteur contenant des identifiants de sens WordNet).

Les nœuds fourmilières possèdent une quantité d'énergie qu'elles peuvent utiliser pour produire des fourmis à chaque itération de l'algorithme.

Les fourmis, partent à l'exploration du graphe de manière pseudo aléatoire. La probabilité de prendre un chemin dépend de la quantité d'énergie sur le nœud, de la concentration de phéromone, et du score entre le nœud (son vecteur de sens) où elle se trouve et sa fourmilière d'origine [5].

Quand une fourmi arrive sur un nœud, elle prélève une quantité d'énergie et a une probabilité dépendant de la quantité d'énergie qu'elle porte de passer en mode retour pour rapporter l'énergie à sa fourmilière.

Lorsqu'une fourmi passe sur un nœud non fourmilière elle va déposer le sens correspondant à sa fourmilière dans le vecteur de sens du nœud ainsi qu'une quantité de phéromone. La phéromone s'évapore en partie à chaque itération.

Si une fourmi arrive sur le nœud d'un autre sens que le sien, il y a une probabilité (dépendant du score avec son sens d'origine) qu'elle construise un "pont" vers sa fourmilière afin d'y revenir directement. Lorsque la fourmi passe par un pont elle dépose également des phéromones, ce qui pourra inciter d'autres fourmis à la suivre.

4. On conserve ainsi la proximité et l'ordre des mots par exemple
5. à l'aide de mesures de similarité locales.

Lorsque de nombreux ponts ont été construits, certains ponts vont se renforcer et d'autres s'évaporer (lorsqu'il n'y aura plus de phéromone) ; cela va mener à une monopolisation des ressources au niveau des fourmilières avec les ponts les plus fréquentés. Les ponts correspondant ainsi à des chemins interprétatifs parmi les combinaisons de sens possibles. À la fin de la simulation les sens qui correspondent aux fourmilières avec le plus d'énergie sont choisis.

5 Critères de choix des algorithmes

Le choix de la mesure de similarité à utiliser dépends d'une part des contraintes sur les ressources lexicales disponibles mais aussi du contexte applicatif : certaines seront plus adaptées car relevant mieux de certains aspects plutôt que d'autres de la similarité sémantique réelle. (Budanitsky et Hirst, 2006) proposent une comparaison empirique de 5 mesures par rapport au jugement humain de manière très détaillée, ce qui peut constituer un élément de choix utile. Plus récemment (Pirró et Euzenat, 2010), entreprennent également de comparer leur mesure à la plupart des mesures classiques.

Quant aux algorithmes globaux, il y a deux aspects à considérer, d'une part l'évaluation de la qualité des solutions, et d'autre part le temps d'exécution de l'algorithme. Les taches de désambiguïsation des campagnes d'évaluation telles que Semeval (anciennement SenseEval), ne sont axées que sur l'évaluation de la qualité par rapport à une désambiguïsation de référence du corpus faite manuellement. Elles fournissent cependant un premier élément de comparaison.

D'une part on peut discuter de la valeur d'une telle évaluation de la qualité dans un système appliqué à un problème réel, et d'autre part de la vitesse d'exécution qui est un facteur très important pour des applications telles que la traduction automatique, surtout si il s'agit de traduction de parole à parole (où il y a un besoin de traitement en temps réel).

(Schwab *et al.*, 2012) ont entrepris de comparer le recuit simulé, l'algorithme génétique ainsi que l'algorithme à colonie de fourmis à la fois en termes de qualité (Semeval 2007 – Tâche 7), mais également en terme de convergence et de vitesse d'exécution en utilisant comme mesure locale Lesk étendu. Ils concluent que les trois algorithmes avec la même mesure de similarité locale offrent des résultats en terme de qualité comparables ; c'est cependant l'algorithme à colonie de fourmis qui s'avère le plus rapide (environ 10 fois plus que le recuit simulé et 100 fois plus que l'algorithme génétique).

6 Conclusions et perspectives de recherche

Nous avons passés en revue les principales méthodes de désambiguïsation lexicale basées sur des connaissances, que ce soit les mesures au niveau local ou les algorithmes au niveau global. D'un point de vue local, les mesures de similarité sémantique sont bien entendu très utiles pour les systèmes de TALN, mais elles jouent également un rôle de plus en plus important pour des applications au web sémantique, et également pour la construction automatisée de ressources lexicales. La recherche se focalise principalement d'une part sur l'hybridation et la combinaison des mesures de similarité, mais également sur la combinaison de sources d'information (multilingues ou non), ou encore des ontologies croisées.

Du point de vue global une possibilité d'amélioration se situe au niveau du temps de convergence et des performances en général. Nous nous intéressons surtout aux combinaisons de mesures de

similarité. Ces combinaisons peuvent se faire en utilisant des mesures différentes pour essayer de capter différents aspects utiles à la désambiguïsation. Par exemple, on peut imaginer utiliser une mesure par catégorie lexicale, utiliser différentes mesures exploitant différentes sources d'information ou adopter une stratégie de vote, ou enfin, au niveau de l'algorithme à colonies de fourmis, utiliser différentes castes de fourmis, chacune utilisant une mesure de similarité différente pour ses déplacements.

Références

BANERJEE, S. et PEDERSEN, T. (2002). An adapted lesk algorithm for word sense disambiguation using wordnet. *In CICLing '02*, pages 136–145, London, UK.

BARZILAY, R. et ELHADAD, M. (1997). Using lexical chains for text summarization. *In In Proceedings of the ACL Workshop on Intelligent Scalable Text Summarization*, pages 10–17.

BRIN, S. et PAGE, L. (1998). The anatomy of a large-scale hypertextual web search engine. *In WWW7*, pages 107–117, Amsterdam.

BUDANITSKY, A. et HIRST, G. (2006). Evaluating wordnet-based measures of lexical semantic relatedness. *Computational Linguistics*, 32(1):13–47.

COLLINS (1998). *Cobuild English Dictionary*. Harper Collins Publishers.

COWIE, J., GUTHRIE, J. et GUTHRIE, L. (1992). Lexical disambiguation using simulated annealing. *In COLING '92*, pages 359–365, Stroudsburg, PA, USA. ACL.

DENEUBOURG, J. L., PASTEELS, J. M. et VERHAEGE, J. C. (1983). Probabilistic behaviour in ants : a strategy of errors ? *Journal of Theoretical Biology*, 105:259–271.

DICE, L. R. (1945). Measures of the amount of ecologic association between species. *Ecology*, 26(3):297–302.

DORIGO, M. (1992). *Optimization, Learning and Natural Algorithms*. Thèse de doctorat, Politecnico di Milano, Italie.

GELBUKH, A., SIDOROV, G. et HAN, S.-Y. (2003). Evolutionary approach to natural language word sense disambiguation through global coherence optimization. *WSEAS Transactions on Communications*, 1(2):11–19.

GELBUKH, A., SIDOROV, G. et HAN, S.-Y. (2005). On some optimization heuristics for lesk-like wsd algorithms. *In NLDB'05*, pages 402–405, Berlin, Heidelberg.

GUINAND, F. et LAFOURCADE, M. (2010). *Artificial ants for Natural Language Processing*, chapitre 20, pages 455–492. Artificial Ants. From Collective Intelligence to Real-life Optimization and Beyond. Monmarché, N. and Guinand, F. and P. Siarry.

HALLIDAY, M. A. et HASAN, R. (1976). *Cohesion in English*. Longman Group Ltd, London, U.K.

HIRST, G. et ST-ONGE, D. D. (1998). Lexical chains as representations of context for the detection and correction of malapropisms. *WordNet : An electronic Lexical Database. C. Fellbaum.*, pages 305–332. Ed. MIT Press.

IDE, N. et VERONIS, J. (1998). Word sense disambiguation : The state of the art. *Computational Linguistics*, 24:1–40.

JIANG, J. J. et CONRATH, D. W. (1997). Semantic Similarity Based on Corpus Statistics and Lexical Taxonomy. *In ROCLING X*.

LEACOCK, C. et CHODOROW, M. (1998). Combining local context and wordnet similarity for word sense identification. *WordNet : An Electronic Lexical Database. C. Fellbaum. Ed. MIT Press. Cambridge. MA*.

LESK, M. (1986). Automatic sense disambiguation using machine readable dictionaries : how to tell a pine cone from an ice cream cone. *In SIGDOC '86*, pages 24–26, New York, NY, USA. ACM.

MIHALCEA, R., TARAU, P. et FIGA, E. (2004). Pagerank on semantic networks, with application to word sense disambiguation. *In COLING '04*, Stroudsburg, PA, USA. ACL.

MILLER, G. A. (1995). Wordnet : a lexical database for english. *Commun. ACM*, 38(11):39–41.

MORRIS, J. et HIRST, G. (1991). Lexical cohesion computed by thesaural relations as an indicator of the structure of text. *Comput. Linguist.*, 17(1):21–48.

NAVIGLI, R. (2009). Word sense disambiguation : A survey. *ACM Comput. Surv.*, 41(2):10 :1–10 :69.

PIRRÓ, G. et EUZENAT, J. (2010). A feature and information theoretic framework for semantic similarity and relatedness. *In ISWC 2010*, volume 6496 de *Lecture Notes in Computer Science*, pages 615–630.

RADA, R., MILI, H., BICKNELL, E. et BLETTNER, M. (1989). Development and application of a metric on semantic nets. *IEEE Transactions on Systems, Man, and Cybernetics*, 19(1):17–30.

RESNIK, P. (1995). Using information content to evaluate semantic similarity in a taxonomy. *In IJCAI'95*, pages 448–453, San Francisco, CA, USA.

ROGERS, D. et TANIMOTO, T. (1960). A computer program for classifying plants. *Science*, 132(3434):1115–1118.

ROGET (1989). *New Roget's Thesaurus*. P.S.I.

SCHWAB, D., GOULIAN, J. et GUILLAUME, N. (2011). Désambiguïsation lexicale par propagation de mesures sémantiques locales par algorithmes à colonies de fourmis. *In TALN*, Montpellier (France).

SCHWAB, D., GOULIAN, J. et TCHECHMEDJIEV, A. (2012). Comparaison théorique et pratique d'algorithmes d'optimisation globaux pour la désambiguïsation lexicale non supervisée. *Traitement Automatique des Langues*, 1(53):37 pages. Soumis à la revue Traitement Automatique des Langues.

SECO, N., VEALE, T. et HAYES, J. (2004). An intrinsic information content metric for semantic similarity in wordnet. *In Proceedings of ECAI'2004*, pages 1089–1090, Valencia, Spain.

SILBER, H. G. et MCCOY, K. F. (2000). Efficient text summarization using lexical chains. *In IUI '00*, pages 252–255, New York, NY, USA. ACM.

TVERSKY, A. (1977). Features of similarity. *Psychological Review*, 84(4):327–352.

WAGNER, C. (2008). Breaking the knowledge acquisition bottleneck through conversational knowledge management. *Information Resources Management Journal*, 19(1):70–83.

WILKS, Y. et STEVENSON, M. (1998). Word sense disambiguation using optimised combinations of knowledge sources. *In COLING '98*, pages 1398–1402, Stroudsburg, PA, USA. ACL.

WU, Z. et PALMER, M. (1994). Verbs semantics and lexical selection. *In Proceedings of the 32nd annual meeting on ACL*, volume 2 de *ACL '94*, pages 133–138, Stroudsburg, PA, USA. Association for Computational Linguistics.

ZIPF, G. K. (1949). *Human Behavior and the Principle of Least Effort*. Addison-Wesley (Reading MA).