TALN 2004, Fès, 19–21 avril 2004

Désambiguïsation de corpus monolingues par des approches de type Lesk

Florentina Vasilescu, Philippe Langlais

RALI/IRO, Université de Montréal
CP. 6128, succursale Centre-ville
Montréal, Québec, H3C CJ7 Canada
{vasilesf,felipe}@iro.umontreal.ca

Résumé - Abstract

Cet article présente une analyse détaillée des facteurs qui déterminent les performances des approches de désambiguïsation dérivées de la méthode de Lesk (1986). Notre étude porte sur une série d'expériences concernant la méthode originelle de Lesk et des variantes que nous avons adaptées aux caractéristiques de WORDNET. Les variantes implémentées ont été évaluées sur le corpus de test de SENSEVAL2, English All Words, ainsi que sur des extraits du corpus SEMCOR. Notre évaluation se base d'un côté, sur le calcul de la précision et du rappel, selon le modèle de SENSEVAL, et d'un autre côté, sur une taxonomie des réponses qui permet de mesurer la prise de risque d'un décideur par rapport à un système de référence.

This paper deals with a detailed analysis of the factors determining the performances of Lesk-based WSD methods. Our study consists in a series of experiments on the original Lesk algorithm and on its variants that we adapted to WORDNET. These methods were evaluated on the test corpus from SENSEVAL2, English All Words, and on excerpts from SEMCOR. The evaluation metrics are based on precision and recall, as in SENSEVAL exercises, and on a new method estimating the risk taken by each variant.

Mots-clefs – Keywords

Désambiguïsation sémantique, algorithme de Lesk, naive Bayes, WORDNET
Word sense desambiguation, Lesk's algorithm, naive Bayes, WORDNET

1 Introduction

La désambiguïsation sémantique d'un texte consiste à déterminer le sens correct des mots de ce texte. Des campagnes d'évaluation comme SENSEVAL sont la preuve du grand intérêt porté au sein de notre communauté à cette tâche (90 équipes ont mentionné leur intérêt à participer à la prochaine campagne SENSEVAL3).

Cet intérêt se traduit par un foisonnement de méthodes et de ressources utilisées, comme par exemple les dictionnaires, les thésaurus ou les lexiques sémantiques électroniques (Lesk, 1986; Banerjee & Pedersen, 2003), les corpus annotés, comportant des étiquettes de sens (voir par exemple (Crestan *et al.*, 2003)), les corpus non annotés (Yarowsky, 1995; Schütze, 1998) ou une combinaison de ces ressources (Stevenson & Wilks, 2001).

Malgré tous ces travaux, nous ne connaissons pas d'application réelle qui tire réellement profit de la désambiguïsation. C'est certes intuitivement une tâche indispensable à la bonne réalisation de toutes les applications qui nécessitent un niveau de compréhension du message d'entrée (la traduction automatique en tête), mais force est de constater que c'est une tâche difficile que nous ne maîtrisons pas complètement.

L'absence de ressources étiquetées de qualité et en grande quantité est une explication souvent avancée pour rendre compte de cet échec. D'autres (*e.g.* (Véronis, 2001)) soulignent que la tâche sur laquelle les différents systèmes se comparent est mal définie. Ils mentionnent en particulier que le niveau de granularité de WORDNET, la ressource utilisée dans la campagne d'évaluation passée, est souvent trop fine pour que même des humains s'accordent sur la bonne étiquette à donner à un mot.

Dans ce contexte, nous avons décidé d'implémenter un algorithme de désambiguïsation simple et tenté de comprendre ses limites autrement qu'en analysant ses performances brutes en terme de précision et de rappel. Le candidat qui nous a semblé le plus intéressant dans cette optique exploratoire est l'algorithme proposé par Lesk (1986) qui consiste à compter le nombre de mots communs entre les définitions d'un mot (généralement trouvées dans un dictionnaire électronique) et les définitions des mots de son contexte. Le sens retenu correspondant à la définition pour laquelle on compte le plus grand nombre de mots communs avec le contexte. Cette idée simple s'est avérée meilleure que bon nombre de techniques plus évoluées dans le cadre de la campagne SENSEVAL1.

Nous avons testé de nombreuses variantes de l'algorithme de Lesk adaptées à WORDNET. Les résultats de ces variantes ont été comparés avec ceux d'une version Naive Bayes (qui peut elle-même être décrite comme une variante de l'algorithme de Lesk). Nous résumons en section 2 les variantes et les facteurs les plus saillants que nous avons étudiés.

La performance obtenue par chaque variante n'étant pas l'objet principal de notre étude, nous avons mis au point une taxonomie des réponses que peuvent faire nos décideurs et qui aide à mieux comprendre leur performance. Nous décrivons cette taxonomie en section 3 et définissons la notion de risque associée à un algorithme de désambiguïsation.

Nous décrivons ensuite en 4 notre protocole expérimental et analysons les performances de chaque variante décrite. Nous discutons finalement nos résultats en section 5.

2 Algorithmes étudiés

Les algorithmes que nous avons implémentés s'appuient tous sur un modèle extrêmement simple décrit en pseudo-code en figure 1. Ce modèle prend en entrée un mot à désambiguïser t ainsi qu'une liste (triée en ordre décroissant de fréquence) de ses sens candidats et produit en sortie le sens sélectionné. Toutes les variantes testées ici diffèrent seulement par le choix associé aux fonctions `Score`, `Description` et `Contexte` décrites dans les sections suivantes.

Entrée:
t, un mot à désambiguïser
$S = \{s_1, \ldots, s_N\}$, les sens candidats ordonnés en ordre décroissant de fréquence

Sortie:
`sens`, l'indice dans S du sens retenu

```
score ← −∞
sens ← 1                                  //choix par défaut du sens le plus fréquent
C ← Contexte(t)                           //contexte du mot cible
for all i ∈ [1,N] do
  D ← Description(s_i)                    //description de s_i extraite de WORDNET
  sup ← 0
  for all w ∈ C do
    W ← Description(w)                    //description de w extraite de WORDNET
    sup ← sup + Score(D,W)                //cumul des superpositions entre D et W
  if sup > score then
    score ← sup
    sens ← i                              //on retient le sens de plus haut score
```

FIG. 1 – Canevas des variantes étudiées. Les trois fonctions dont dépend ce modèle sont décrites à même le texte.

2.1 Définition du contexte

La première des fonctions dont dépend le modèle de la figure 1 — `Contexte(t)` — définit l'ensemble des mots qui vont servir à la désambiguïsation de t. Nous avons testé deux implémentations de cette fonction. La première — celle utilisée par défaut — consiste à retourner l'ensemble des mots pleins centrés autour du mot t. Nous rapportons les résultats de nos variantes en section 4 pour des longueurs de contexte de ± 2 (les deux mots pleins directement à gauche et à droite de t), ± 3, ± 8, ± 10 et ± 25 mots. Notons qu'Audibert (2003) suggère que choisir un contexte symétrique n'est pas optimal dans le cas des verbes. Il montre en effet que l'information servant à les désambiguïser a tendance à se trouver dans les compléments d'objets, et donc plutôt à droite des verbes. L'auteur suggère pour cela d'utiliser un contexte $< -2, +4 >$. Dans les mêmes actes, Crestan et al. (2003) montrent qu'il est bénéfique — du moins pour certaines classes syntaxiques de mots — de mettre en œuvre une procédure automatique de sélection du contexte.

La seconde implémentation de cette fonction, dénotée CL, consiste à extraire du contexte d'oc-

currence de t les mots constitutifs de ce que nous appelons après Hirst et St-Onge (1998) sa *chaîne lexicale*. Dans une étude sur la correction des *malapropismes* (confusion de deux mots comportant la même prononciation ou des formes orthographiques très semblables mais des sens différents), les auteurs s'appuient sur l'idée que pour rendre un discours cohérent, les mots cooccurrant dans un même contexte sont reliés entre eux par des relations de cohésion, formant des enchaînements logiques qu'ils baptisent chaînes lexicales.

Nous avons adapté cette idée à la désambiguïsation sémantique, en considérant que la levée de l'ambiguïté d'un mot peut se faire en déterminant la chaîne lexicale de ce mot. Seuls les mots de cette chaîne sont alors retenus dans le contexte. Notre implémentation utilise les relations de synonymie et d'hyperonymie de WORDNET ainsi qu'une mesure de similarité entre deux ensembles (nous avons utilisé la formule de Jaccard) pour tester l'appartenance d'un mot du contexte de t à sa chaîne lexicale. Son principe consiste à associer à chaque mot w du contexte l'ensemble $E(w)$ des mots des *synsets* rencontrés en suivant les liens (jusqu'à la racine) d'hyperonymie et de synonymie définis dans WORDNET pour chaque sens de w. On décide alors que w appartient à la chaîne lexicale de t si le score de Jaccard pour les deux ensembles $E(w)$ et $E(t)$ est supérieur à un seuil fixé empiriquement. Le résultat de ce processus est illustré en figure 2.

Committee approval of Gov._Price_Daniel's "abandoned property" act seemed certain Thursday despite the adamant protests of Texas bankers. Daniel personally led the fight for the measure, which he had watered_down considerably since its rejection by two previous **Legislatures**, in a **public** hearing before the **House_Committee_on_Revenue_and_Taxation**. Under **committee** rules, it went automatically to a **subcommittee** for one week.

– $E(committee)$ = {committee, commission, citizens, administrative-unit, administrative-body, organization, social-group, group, grouping}
– $E(legislature)$ = {legislature, legislative-assembly, general-assembly, law-makers, assembly, gathering, assemblage, social-group, group, grouping}

FIG. 2 – Illustration d'une chaîne lexicale. Les mots en gras forment la chaîne lexicale du mot *committee*. $E(committee)$ et $E(legislature)$ sont les ensembles de mots obtenus en suivant les relations de synonymie et d'hyperonymie dans WORDNET.

2.2 Description associée à un mot

L'algorithme de la figure 1 applique une fonction de score à la représentation `Description(w)` d'un mot w du contexte de t ainsi qu'à celle d'un sens particulier s_t de t: $Description(s_t)$. Dans tous nos tests, l'entité descriptive d'un sens est représentée par un sac de mots, c'est-à-dire un ensemble de mots dont l'ordre et la dépendance sont ignorés. Cet ensemble contient seulement des mots pleins (noms, verbes, adjectifs ou adverbes) dans leur forme canonique (lemme) et l'entité descriptive associée à un mot est l'union des entités descriptives de chacun des sens de ce mot (`Description(w)` $= \bigcup_{s \in Sens(w)}$ `Description(s)` ; où $Sens(w)$ est l'ensemble des sens de w selon WORDNET).

Nous avons étudié trois variantes de la représentation d'un sens. La première, dénotée DEF,

consiste à ne conserver que les mots pleins de la définition associée au sens selon WORDNET[1]. La deuxième, notée REL, consiste à regrouper les mots des synsets parcourus en suivant dans WORDNET les relations de synonymie et d'hyperonymie (c'est comme cela qu'a été obtenue la représentation $E(committee)$ dans l'exemple de la figure 2). Une troisième variante, DEF+REL, consiste à faire l'union des deux représentations précédentes.

Une autre variante de la fonction `Description` permet d'implémenter la variante simplifiée de Lesk proposée par Kilgarriff et Rosenzweig (2000). Cette variante consiste à comptabiliser les intersections entre l'entité descriptive d'un sens candidat et les mots du contexte de t (et non plus leur définition). Dans ce cas, la description associée à un mot du contexte est très simple (`Description(w) = {w}`); la description d'un sens candidat n'est quant à elle pas changée (DEF, REL, ou DEF+REL).

2.3 Fonction de score

Nous avons testé de nombreuses variantes de la fonction de score `Score`(E_1,E_2) entre deux entités descriptives E_1 et E_2 dont les détails et les performances peuvent être lus dans (Vasilescu, 2003). Nous décrivons ici les classes de variantes les plus saillantes qui sont toutes des fonctions cumulatives du score de chaque intersection entre E_1 et E_2.

La variante la plus simple, désignée par LESK dans la suite, consiste à donner à chaque intersection le score unitaire. C'est le score qui correspond à l'algorithme de Lesk.

Une deuxième classe de variantes dénommée PONDÉRÉ suit la suggestion faite par (Lesk, 1986) que le score devrait tenir compte de la taille de l'entrée du dictionnaire pour un sens donné afin d'éviter que les descriptions trop longues ne dominent le processus de prise de décision (plus une description est longue et plus les intersections sont probables). Le score associé à une intersection entre deux mots est donc normalisé par l'inverse du logarithme de la taille (comptée en mots) de la description du sens candidat. Nous avons étudié d'autres mécanismes de pondération tenant par exemple compte de la distance du mot du contexte au mot cible, ou encore de la fréquence d'occurrence dans la langue du mot du contexte sans obtenir de gain lors de la désambiguïsation.

La troisième fonction de score utilisée, désignée par BAYES, est spécifique à notre implémentation naive Bayes qui s'inscrit également dans le canevas de la figure 1. Une telle approche sélectionne en effet parmi les sens candidats s du mot cible celui qui maximise la quantité $p(s|Contexte(t))$ en faisant comme hypothèse que tous les mots du contexte sont indépendants. Précisément, notre fonction de score est:

$$\log p(s) + \sum_{w \in Contexte(t)} \log \left(\lambda p(w|s) + (1-\lambda)p(w)\right)$$

où les trois distributions $p(s)$, $p(w|s)$ et $p(w)$ sont déterminées par fréquence relative à partir du corpus SEMCOR décrit brièvement dans la section 4.1. Le lissage de $p(w|s)$ par un modèle unigramme $p(w)$ est ici nécessaire compte tenu de la forme très piquée des distributions conditionnelles (comme nous l'avons mentionné en introduction, les corpus étiquetés sont de petite taille). Ce lissage est contrôlé par un unique paramètre λ fixé à 0.95 dans nos expériences.

[1]Cette définition peut contenir également des exemples d'usage.

3 Métriques d'évaluation

À l'instar des campagnes SENSEVAL, nous mesurons les performances de nos différents décideurs à l'aide des mesures classiques de *précision* et de *rappel*. La précision (resp. le rappel) est le ratio du nombre de réponses correctes fournies par le système sur le nombre de décisions faites (resp. à prendre). Ces mesures ne permettent qu'indirectement d'apprécier le mérite de chaque décideur (au delà du fait qu'un bon décideur est celui pour lequel on mesure une précision et un rappel élevé). Aussi proposons-nous une taxonomie des réponses faites par nos algorithmes qui permet de comparer un décideur à un système de référence: ici le système BASE qui retourne toujours le sens candidat le plus fréquent selon WORDNET.

Cette taxonomie fait intervenir deux caractéristiques propres à une réponse, à savoir sa correction (C=correcte, $\overline{C}$ = incorrecte) et son effectivité (E=effective, $\overline{E}$ = non effective) ainsi que deux caractéristiques supplémentaires liées à la réponse du système de référence (BASE) qui peut être juste (B) ou fausse ($\overline{B}$) tout en étant égale ($=$) ou pas ($\neq$) à la réponse faite par le système testé. Nous qualifions une réponse d'*effective* lorsqu'au moins une intersection a été observée entre les représentations du contexte et la représentation du sens choisi (nous rappelons qu'en cas de non décision, le sens le plus fréquent selon WORDNET est toujours sélectionné). En prenant comme système de référence le système BASE, nous obtenons une combinaison de 7 classes qui sont représentées en figure 3. Nous pouvons alors définir deux types de risques.

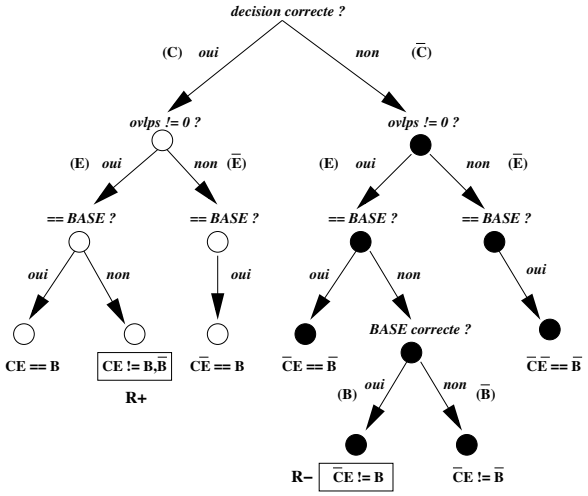

FIG. 3 – Taxonomie des réponses faites par un décideur. La classe $CE = B$ caractérise par exemple les réponses effectives correctes qui sont identiques au système de référence ; alors que la classe $CE \neq \overline{B}$ désigne les réponses effectives correctes différentes de la réponse du système de base qui elle est fausse. $ovlps$ désigne le nombre de superpositions considérées lors de la prise de décision.

Le *risque positif* ($R+$) est donné par le nombre ($CE \neq B, \overline{B}$) de décisions effectives correctes, différentes des décisions de BASE. Le *risque négatif* ($R-$) est le nombre ($\overline{C}E \neq B$) de décisions effectives incorrectes, pour lesquelles le système de référence était quant à lui correct. Ces deux quantités sont normalisées par le nombre total de décisions faites. La différence entre ces deux mesures détermine ce que nous appelons le *gain* par rapport aux performances du système de référence.

4 Expériences

4.1 Protocole

Nous avons utilisé la version 1.7.1 de WORDNET pour obtenir les différents sens possibles de chaque mot ainsi que les définitions et les relations associées à chaque sens. Les mots à désambiguïser qui n'étaient pas présents dans WORDNET (0.8% des instances) sont comptabilisés comme des erreurs du décideur testé. C'est également les informations fournies par WORDNET qui nous permettent de classer les différents sens candidats entre eux (nous nous basons sur les champs *sense_number* et *tag_cnt* de la table d'index de WORDNET).

Nous avons de plus utilisé la version 1.7.1 du corpus SEMCOR pour entraîner nos versions naive Bayes. Dans le but de vérifier la stabilité de nos observations, nous avons testé nos décideurs sur le corpus de test de la campagne SENSEVAL2 ainsi que sur différents jeux de test que nous avons créés à partir du corpus SEMCOR (les approches naive Bayes entraînées sur ce même corpus n'ont cependant pas été testées sur ces jeux de test SEMCOR). Les résultats que nous rapportons ici sont ceux obtenus sur le premier corpus (nous laissons le soin au lecteur intéressé de lire les différences inter-corpus dans (Vasilescu, 2003)), car c'est le corpus qui sert de point de comparaison dans la plupart des études en désambiguïsation (2473 mots cibles). Nous veillons cependant dans notre discussion à ne dégager que les tendances observées sur l'ensemble de nos jeux de test.

Chaque décideur avait à charge de désambiguïser l'ensemble des mots pleins du texte d'entrée (les mots étiquetés *head* dans le corpus SENSEVAL2), ce que l'on désigne habituellement par la piste *English All Words*.

4.2 Performances

4.2.1 Comparaison des différents décideurs

Le tableau 1 présente la précision et le rappel des décideurs décrits dans la section 2 dans leur version DEF. Plusieurs tendances se dégagent de ce tableau. En premier lieu, il convient de noter que l'algorithme de Lesk dans sa formulation originelle (LESK) donne les moins bons résultats. En fait, les résultats de ce décideur sont de loin inférieurs à ceux obtenus en prenant le sens le plus fréquent. Cette observation est cependant cohérente avec celles de Litkowski (2002) où l'auteur analyse les différents facteurs responsables des performances de son système à SENSEVAL2 (CL Research - DIMAP, 29.3% de précision et rappel English lexical sample[2]). Il fait en particulier l'observation que seulement 30% des instances à désambiguïser tiraient profit de l'information de type Lesk (définitions + exemples).

Il est également clair que la version simplifiée (SLESK) de cet algorithme donne de meilleurs résultats. Rappelons que la version simplifiée consiste a compter les intersections entre la description associée à chaque sens candidat et les mots du contexte eux-même (et non leur description).

Nous constatons que de pondérer le score par l'inverse de la longueur de la description, tel que le suggérait Lesk (1986) ne s'avère pas une stratégie payante, et ce aussi bien pour les

[2]Voir le site `www.cs.unt.edu/~rada/senseval`.

	P	±2	R	P	±3	R	P	±8	R	P	±10	R	P	±25	R
LESK	42.64		42.26	42.96		42.58	43.21		42.82	43.29		42.90	42.39		42.01
+ PONDÉRÉ	39.29		38.94	39.41		39.06	41.21		40.84	40.76		40.40	41.49		41.12
+ CL	58.38		57.86	58.22		57.70	56.18		55.68	55.65		55.16	53.90		53.42

	P	±2	R	P	±3	R	P	±8	R	P	±10	R	P	±25	R
SLESK	58.18		57.66	57.20		56.69	54.67		54.19	53.28		52.81	50.47		50.02
+ PONDÉRÉ	56.67		56.17	55.49		54.99	51.08		50.63	49.25		48.81	44.39		44.00
+ CL	59.08		58.55	59.12		58.59	58.43		57.91	58.26		57.74	57.41		56.89

	P	±2	R	P	±3	R	P	±8	R	P	±10	R	P	±25	R
BAYES	57.60		57.30	58.00		57.70	56.80		56.60	57.60		57.30	58.50		58.30

TAB. 1 – Précision et rappel des différents décideurs décrits dans leur variante DEF en fonction de la taille du contexte. Le système BASE obtient une précision de 57.99 et un rappel de 57.62 (performance qui ne dépend d'aucun paramètre).

variantes LESK que SLESK. En revanche, le filtrage du contexte à l'aide des chaînes lexicales permet d'augmenter les performances de toutes les variantes testées. Ceci semble attester qu'il n'est pas souhaitable de considérer tous les mots lors d'une prise de décision. Les améliorations apportées par le filtre CL aux variantes LESK sont particulièrement marquées. Il est toutefois surprenant qu'un tel filtre porte ses fruits dans des contextes très resserrés autour du mot cible (± 2). L'analyse que nous fournissons plus loin propose une explication des performances de cette variante.

Nous pouvons également observer qu'à l'exception des variantes LESK et BAYES, augmenter la taille du contexte entraîne une décroissance des performances, ce qui appuie l'importance de la sélection d'un bon contexte de désambiguïsation. Notons enfin que la meilleure des variantes testées ici ne dépasse pas de façon très marquée le système BASE.

4.2.2 Choix de la description associée à un sens

Il est difficile de dégager des conclusions claires quant à l'influence du choix de la fonction `Description`. Cependant la tendance la plus marquée (et ce pour toutes les configurations testées) est que pour un contexte très court (± 2), il est préférable de considérer les mots de la définition des sens (DEF) que les relations (REL). Pour des contextes plus grands, il semble en revanche que les relations donnent de meilleurs résultats. Il est cependant difficile d'interpréter cette observation sans recourir à une analyse plus poussée.

4.2.3 Analyse des réponses

La taxonomie que nous avons présentée en section 3 nous permet de comprendre davantage les différentes variantes testées. Le tableau 2 reporte le risque positif ($CE \neq B, \overline{B}$) et négatif ($\overline{C}E \neq B$) des différentes variantes SLESK étudiées. Nous pouvons observer que, à l'exception des variantes CL, les décideurs prennent plus de risque négatif que positif, et ce d'autant plus que le contexte est grand.

Pour toutes les variantes testées, le taux des réponses correctes différentes de BASE est très petit.

	±2		±3		±8		±10		±25	
	R+	R-	R+	R-	R+	R-	R+	R-	R+	R-
SLESK	3.5	3.3	*3.9*	*4.7*	*6.0*	*9.3*	*6.5*	*11.2*	*7.8*	*15.3*
+ PONDÉRÉ	*3.5*	*4.8*	*3.9*	*6.4*	*5.9*	*12.8*	*6.4*	*15.2*	*7.8*	*21.3*
+ CL	1.1	0.2	1.2	0.2	1.7	1.3	1.7	1.5	*1.9*	*2.5*

TAB. 2 – Risque positif (R+) et négatif (R-) des différentes variantes SLESK. Les mesures rapportées en italique indiquent des gains négatifs par rapport au système BASE.

La majorité des réponses correctes coïncide en fait avec les réponses correctes de BASE ; soit qu'elles sont prises par défaut en l'absence de superposition ($C\overline{E} = B$, le cas le plus fréquent), soit qu'elles sont produites par des décisions effectives ($CE = B$). Dans le cas des variantes CL, les réponses correctes sont majoritairement des réponses non effectives: cette variante doit ses performances à une stratégie silencieuse (quelques décisions effectives prises à bon escient, la plupart étant des décisions non effectives). Les autres variantes prennent quant à elles plus de risque et la proportion de bonnes réponses effectives est donc plus grande. Elles coïncident cependant majoritairement avec le choix du sens le plus fréquent.

4.2.4 Filtrage par étiquetage morpho-syntaxique

Nous avons étudié l'impact du filtrage des sens candidats à l'aide de l'étiquette morpho-syntaxique connue (APOS) ou estimée (RALI) du mot cible. Dans le second cas, nous avons fait usage d'un étiqueteur développé au RALI ; un modèle markovien d'ordre 3 entraîné sur le corpus des débats parlementaires canadiens. Soulignons que ce corpus est par nature très différent du corpus de test de SENSEVAL2. L'étiquette était utilisée comme filtre. Par exemple selon WORDNET, le mot anglais *house* contient 12 sens en tant que nom et seulement 2 sens en tant que verbe. Le fait de savoir (APOS) ou de croire (RALI) qu'on est par exemple en présence d'un verbe permet de ne considérer que 2 sens candidats. Les performances de cette approche sont rapportées dans le tableau 3. Comme on peut le constater, la prise en compte de l'information morpho-syntaxique (estimée ou connue) améliore les performances de l'approche de base (sens le plus fréquent). Nous n'observons cependant pas de gain lorsque l'approche de base bénéficie également de cette information.

	APOS	P	R	RALI	P	R	P	R
SLESK+ CL		61.9	61.3		60.5	59.9	59.1	58.6
BASE		61.9	61.3		60.4	59.9	57.9	57.6

TAB. 3 – Précision et rappel de la meilleure variante testée lorsque la catégorie morpho-syntaxique du mot à désambiguïser est connue (APOS) ou estimée (RALI). La dernière colonne rappelle les performances obtenues sans ce filtrage.

5 Conclusion

Nous avons fait l'étude de différentes variantes de l'algorithme de Lesk et tenté d'analyser leur performance grâce à une taxonomie des réponses que nous avons décrite. Nos expériences

ont montré qu'en général les performances diminuent avec l'élargissement du contexte, les meilleures performances étant enregistrées par des fenêtres de 4 à 6 mots pleins autour du mot cible. Nous avons observé que filtrer les mots du contexte (ici par les chaînes lexicales) était bénéfique (le décideur prend moins de risque négatif).

La catégorie grammaticale peut de plus agir comme un filtre qui réduit le nombre de sens candidats possibles d'un mot cible. Les performances obtenues en se servant d'une étiquette morpho-syntaxique estimée permettent de dépasser le meilleur des décideurs qui n'utilise pas cette information. Rappelons que dans le cadre de SENSEVAL2 (English All Words), le meilleur système supervisé a obtenu une performance (précision et rappel) de 69% alors que le meilleur système non supervisé obtenait une précision de 57.5% et un rappel de 56.9%.

Les tentatives ici décrites suggèrent que les mots des définitions, des exemples d'usage et des relations ne sont pas suffisants pour une bonne désambiguïsation. Ceci rejoint les observations de Véronis (2001) qui souligne que les définitions de WORDNET (et des dictionnaires en général) ne fournissent pas toujours l'information nécessaire à la désambiguïsation. L'ajout d'une information de type syntaxique ou pragmatique reliée à l'usage des mots dans des contextes réels semble nécessaire à ce type de tâche.

Références

AUDIBERT L. (2003). Étude des critères de désambigïsation sémantique automatique: résultats sur les cooccurrences. In *10e conférence TALN*, p. 35–44, Batz-sur-mer, France.

BANERJEE S. & PEDERSEN T. (2003). Extended gloss overlaps as a measure of semantic relatedness. In *Eighteenth International Conference on Artificial Intelligence (IJCAI-03)*, p. 805–810, Acapulco, Mexico.

CRESTAN E., EL-BÈZE M. & DE LOUPY C. (2003). Peut-on trouver la taille de contexte optimale en désambiguïsation sémantique ? In *10e conférence TALN*, p. 85–94, Batz-sur-mer, France.

HIRST G. & ST-ONGE D. (1998). Lexical chains as representation of context for the detection and correction of malapropisms. In C. FELLBAUM, Ed., *WordNet: An electronic lexical database and some of its applications*, p. 305–331. Cambrige, MA: The MIT Press.

KILGARRIFF A. & ROSENZWEIG J. (2000). Framework and results for English SENSEVAL. In *Computers and the Humanities*, volume 34, p. 15–48. Kluwer.

LESK M. (1986). Automatic sense disambiguation using machine readable dictionaries: How to tell a pine cone from an ice cream cone. In *The Fifth International Conference on Systems Documentation, ACM SIGDOC*.

LITKOWSKI K. (2002). Sense information for disambiguation: Confluence of supervised and unsupervised methods. In *SIGLEX/Senseval Workshop on Word Sense Disambiguation: Recent Successes and Future Directions*, Philadelphia.

SCHÜTZE H. (1998). Automatic word sense discrimination. *Computational Linguistics*, **24**(1), 97–123.

STEVENSON M. & WILKS Y. (2001). The interaction of knowledge sources in word sense disambiguation. *Computational Linguistics*, **27**(3), 321–351.

VASILESCU F. (2003). Désambiguïsation de corpus monolingues par des approches de type Lesk. Master's thesis, Université de Montréal.

VÉRONIS J. (2001). Sense tagging: does it make sense ? In *The Corpus Linguistics Conference*, Lancaster, UK.

YAROWSKY D. (1995). Unsupervised word sense disambiguation rivaling supervised methods. In *33rd Meeting of the Association for Computational Linguistics*, p. 189–196.