Publié dans Revue Tranel (Travaux neuchâtelois de linguistique) 21, 247-263, 1994
qui doit être utilisée pour toute référence à ce travail

DELENE: un désambiguïsateur lexical neuronal pour textes en langue seconde ◊

Franck Bodmer

Résumé

Dans cet article, nous présentons un **d**ésambiguïsateur **l**exical **ne**uronal nommé DELENE que nous avons intégré dans le correcteur grammatical en langue seconde ARCTA développé au Laboratoire TLP de l'Université de Neuchâtel. DELENE est capable d'apprendre à désambiguïser plus de 98% des ambiguïtés d'un corpus L2 connu (corpus de textes anglais d'apprenants) en n'utilisant pour toute information que la syntaxe locale. En favorisant l'apprentissage de connaissances généralisables à des textes inconnus, DELENE obtient plus de 92% sur un texte de test L2. Les meilleurs résultats ont été atteints en élargissant un ensemble de 14 catégories grammaticales de base à 23. Une version *hybride* de DELENE, utilisant la morphologie en plus de la syntaxe locale, nous a permis de traiter le cas plus difficile des mots *inconnus* avec un taux d'attribution de la catégorie correcte de 88% sur le même corpus d'apprentissage L2, et de 82% sur le corpus de test.

1. Introduction

Dans cet article, nous présentons un **d**ésambiguïsateur **l**exical **ne**uronal nommé DELENE que nous avons intégré dans le correcteur grammatical en langue seconde ARCTA développé au Laboratoire TLP de l'Université de Neuchâtel[1].

Etant donné que le terme d'ambiguïté lexicale est défini de façon très variée dans la littérature spécialisée, commençons par préciser que nous n'aborderons ici que le phénomène de l'homographie, c'est-à-dire des mots pouvant avoir plusieurs fonctions (ou catégories) syntaxiques.

La désambiguïsation lexicale a fait son entrée dans l'informatique linguistique dans les années soixante, quand les premiers corpus sur support informatique ont été constitués. Les premiers désambiguïsateurs

◊ Cette étude a pu être entreprise et menée à bien grâce à un subside de la CERS (no. 2054.2).

[1] Voir l'article de C. Tschumi, F. Bodmer, E. Cornu, F. Grosjean, L. Grosjean, N. Kübler & C. Tschichold (ce numéro).

ont été très coûteux en temps à réaliser et ne donnent probablement de bons résultats que sur le corpus pour lequel ils ont été créés. Les nouveaux désambiguïsateurs des années quatre-vingt sont caractérisés par des algorithmes ayant pour but de diminuer l'effort humain pour les mettre en œuvre, d'automatiser l'apprentissage, de faciliter leur adaptation à de nouveaux textes et d'abaisser le temps de calcul. Leurs performances plafonnent autour de 95% de mots correctement étiquetés (= étiquetage et désambiguïsation). DELENE est, dans cette lignée, notre contribution dans le domaine neuronal.

Depuis quelques années seulement, le traitement du langage naturel (TLN) a trouvé dans les réseaux neuronaux (RN) un champ d'expérimentation nouveau. Force est de reconnaître que si, jusqu'à présent, peu de systèmes de TLN connexionnistes ont vu le jour, par contre à peu près tous les niveaux d'analyse de la langue abordés par l'informatique linguistique classique ont fait l'objet de recherches dans les milieux connexionnistes. Voici quelques exemples classés par niveaux d'analyse et pôles d'intérêt :

niveau morphologique : étiquetage des mots : Elenius & Carlson (1989) ;

niveau syntaxique : les grammaires libres de contexte : Fanty (1986) ; théorie du gouvernement et liage : Rager & Berg (1990) ; autres analyseurs connexionnistes : Nakagawa & Tatsunori (1988), Jain (1991), Faisal & Kwasny (1990) ; désambiguïsation lexicale : Anderson & Benello (1989), Eizirik & al.(1993) ;

niveau sémantique : assignation de rôles thématiques : Miikkulainen & Dyer (1991), St. John & McClelland (1990) ; scripts : Miikkulainen & Dyer (1991) ; désambiguïsation lexicale (sémantique) : Eizirik & al. (1993) ;

problème de représentation : lexique connexionniste : Miikkulainen & Dyer (1991) ; représentation sémantique : Veronis & Ide (1990), Scholtes (1991) ; représentation de connaissances syntaxiques : Elman (1991) ; représentation en général : Van Gelder (1989), Pollack (1990), Hinton (1990), Smolensky (1990) ;

traitement séquentiel : Gasser & Dyer (1988), Elman (1989, 1991) ;

traitement symbolique : Derthick (1990), Touretzky (1990), Smolensky (1990) ;

extraction de connaissances linguistiques : Crucianu & Memmi (1992), Elman (1989).

Pourquoi avoir choisi les réseaux neuronaux pour la désambiguïsation lexicale? L'approche neuronale, à travers l'apprentissage automatique, semble permettre d'éviter la construction manuelle des règles de désambiguïsation et n'utilise en mémoire que la place pour stocker la matrice des poids. Un inconvénient réside dans le temps de calcul lors du processus de désambiguïsation ; il s'agit donc de trouver un compromis entre des performances suffisantes et un temps d'attente raisonnable. La disponibilité d'un grand corpus de textes anglais déjà étiquetés, le corpus Brown, avec plus d'un million de mots, nous a poussés vers une méthode avec apprentissage et évaluation automatique. Malgré le fait que des connaissances linguistiques diverses soient impliquées dans le phénomène de l'ambiguïté lexicale, il semble que le mécanisme de la désambiguïsation repose en grande partie sur un comportement statistique au niveau des catégories syntaxiques, ce qui, bien sûr, ouvre la porte aux méthodes neuronales.

Le contenu de cet article est le suivant : Après un rapide aperçu du phénomène de l'ambiguïté lexicale et des travaux qui ont déjà été réalisés dans ce domaine (section 2), nous expliquerons le fonctionnement de DELENE (section 3), nous présenterons les résultats obtenus (section 4) et nous terminerons par une version *hybride* de DELENE permettant d'ajouter des connaissances morphologiques pour traiter les mots inconnus (section 5).

2. La désambiguïsation lexicale

La désambiguïsation lexicale est le processus de base de tout système de TLN devant traiter des textes de façon intelligente. De ce pré-traitement dépend le succès que l'on obtiendra dans les étapes ultérieures. Le travail effectué lors du projet ARCTA a bien démontré qu'une

désambiguïsation correcte à 92% n'est pas suffisante pour la détection d'erreurs.

L'ambiguïté lexicale n'est pas un problème avec des données fixes. Selon la richesse du lexique qu'un système utilise, 'the' sera non-ambigu (Det) ou ambigu (Det et Adv). A des sens rares ou provenant d'un domaine particulier/spécialisé comme par exemple *to people* ou *to japan*, peuvent s'ajouter des problèmes de définition et de classement quand on se procure un lexique externe : *my* dans un lexique sera adjectif possessif, mais il peut être pronom dans un autre. Des problèmes de terminologie de ce genre peuvent créer des difficultés à un désambiguïsateur.

Les connaissances linguistiques nécessaires à la désambiguïsation lexicale sont très variées. Les plus utilisées sont la syntaxe locale et la fréquence des catégories d'une unité lexicale. Elles permettent de désambiguïser la majeure partie des cas. Des cas plus difficiles nécessitent l'aide de la syntaxe globale et/ou de la sémantique (ex.: *I came before [CONJ] you did*, et *I came before [PREP] you*). On peut faciliter le processus de désambiguïsation en pré-filtrant des expressions figées ou semi-figées d'une part, et en utilisant un nombre plus ou moins important de catégories syntaxiques d'autre part (de 30 à 150 dans les systèmes connus). Enfin, le style du texte, sa fonction (titre, énumération, etc.) et les compétences de la personne qui rédige sont autant de facteurs auxquels un désambiguïsateur doit pouvoir s'adapter.

Etant donné que l'ambiguïté n'est pas un phénomène isolé, mais que l'on rencontre des *séquences* de mots ambigus, les systèmes proposés abordent la désambiguïsation de trois façons différentes : soit 1) par la désambiguïsation au mot par mot, en avançant dans la phrase de gauche à droite (ou de droite à gauche), depuis un terrain "sûr" (non-ambigu ou désambiguïsé) vers un terrain ambigu (ex. TAGGIT de Greene & Rubin (1971)), soit 2) en procédant par paquet, c'est-à-dire en cherchant à désambiguïser une séquence de mots ambigus délimitée par des mots non-ambigus (ex. CGC de Klein & Simmons (1963), CLAWS de Garside & al. (1987) ; VOLSUNGA de DeRose (1988) ; la méthode stochastique de Church (1988) ; le modèle de Markov de Cutting & al.(1992)), soit 3) en commençant par attribuer une catégorie préférentielle à chaque mot et en modifiant certaines catégories par un ensemble de règles jusqu'à

obtention d'un état stable (ex. DILEMMA de Martin & al.(1988) et de Paulussen (1992) ; le système incrémental de Brill (1992)). Parmi les inconvénients majeurs de ces systèmes, il faut citer, pour la première approche, qu'ils ne peuvent s'appuyer que sur une partie du contexte, et pour la deuxième, qu'ils sont confrontés à un problème d'explosion combinatoire.

La plupart des systèmes cités sont des *taggers*, c'est-à-dire des étiqueteurs-désambiguïsateurs. Les évaluations qui sont publiées par leurs auteurs (et qui tournent généralement autour de 95%) indiquent pour cette raison le pourcentage de mots étiquetés et désambiguïsés correctement, et non, comme dans notre cas, le pourcentage de mots désambiguïsés correctement. Quant aux erreurs qui restent, il y a de fortes chances qu'elles soient dues à des difficultés d'ordre sémantique et à quelques cas de syntaxe globale et qu'elles ne puissent être surmontées pendant quelques années encore.

3. Le désambiguïsateur DELENE

Dans cette section, nous allons décrire l'approche neuronale que nous avons utilisée pour construire le désambiguïsateur DELENE et expliquer son fonctionnement. En particulier, nous nous pencherons rapidement sur les phases d'apprentissage et de reconnaissance (ici, la reconnaissance est la phase de désambiguïsation) qui sont des caractéristiques des méthodes neuronales. Une bonne introduction aux méthodes neuronales est proposée par Hertz & al. (1991).

3.1 Un perceptron multicouches pour la désambiguïsation

Parmi les différentes familles de réseaux neuronaux, nous avons choisi le perceptron multicouches (Rumelhart & al. 1986) pour effectuer la tâche de désambiguïsation. Celui-ci est composé de :

- une couche d'entrée d'unités appelées neurones (d'après le modèle biologique simplifié dont il découle) dont le nombre est déterminé de sorte que le problème soumis au réseau puisse être codé ;

- une couche de sortie qui code la réponse du réseau ;

- au moins une couche intermédiaire (dite cachée).

Chaque neurone d'une couche donnée est relié aux neurones de la couche suivante par un lien pondéré. D'autre part, son activité est une fonction de l'activité des neurones reliés à son entrée et des poids qui les relient. Pour pouvoir être utilisé sur une tâche donnée, le perceptron doit passer par une phase d'apprentissage au cours de laquelle on lui présente des échantillons (ici, des homographes en contexte). Le réseau modifie ses poids (= il apprend) sur la base de la réponse correcte qu'on lui présente (ici, la catégorie correcte de l'homographe). Si le nombre d'échantillons est suffisamment grand, le réseau parvient à généraliser ses connaissances, ce qui le rend utilisable sur des échantillons qu'il n'a pas encore vus.

3.2 Le fonctionnement de DELENE

DELENE est basé sur le modèle d'Anderson & Benello (1989), que nous avons modifié pour nos besoins. La désambiguïsation se fait sur une fenêtre de quatre mots, le mot ambigu ayant l'index i. La couche d'entrée est donc composée de quatre blocs et chaque unité à l'intérieur d'un bloc représente une catégorie syntaxique. Dans CATS(i), on active les catégories du mot ambigu, dans CAT(i-1) et CAT(i-2), la catégorie des mots (désambiguïsés) de gauche, dans CATS(i+1) la ou les catégories du mot de droite.

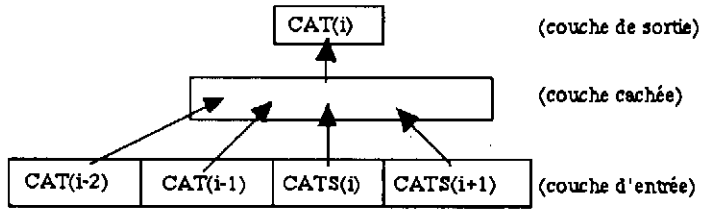

Fig. 1 : schéma de fonctionnement de DELENE

L'apprentissage a pour but d'entraîner la sortie CAT(i) à proposer la distribution probabilistique des catégories du mot i dans son contexte. En mode de désambiguïsation, on choisit dans CAT(i) la catégorie du mot i ayant la plus haute activation.

DELENE est un pur modèle de désambiguïsation lexicale. Il a l'avantage d'utiliser pleinement les possibilités d'un réseau neuronal dans

la mesure où : 1) le phénomène de l'ambiguïté (ambiguïté du mot et de son contexte) peut directement être représenté, et 2) il apprend la tâche de désambiguïsation directement à travers des exemples. L'apprentissage peut se faire automatiquement en extrayant et présentant au réseau tous les cas d'ambiguïté rencontrés dans un corpus de phrases où les catégories syntaxiques sont connues et déjà désambiguïsées.

4. Les résultats

Nous présentons maintenant les résultats obtenus sous les différents aspects qui nous ont semblé intéressants.

4.1 Premiers résultats

Comme pour les méthodes statistiques et probabilistes, les méthodes neuronales demandent de grandes quantités de textes pour donner de bons résultats. Sachant par ailleurs que diverses méthodes de désambiguïsation lexicale avaient été développées sur la base du corpus Brown (la raison étant que ce corpus d'un million de mots est annoté et désambiguïsé), nous avons commencé par faire apprendre à DELENE des extraits du Brown. Pour cette première étape, nous avons utilisé les 14 catégories du prototype ARCTA (voir section 4.3). Les résultats indiquèrent que le réseau pouvait apprendre à désambiguïser, mais qu'il restait bloqué à un seuil insuffisant. Nous avons évalué DELENE sur un fichier d'apprentissage et un fichier de test et obtenu les résultats suivants, où le pourcentage désigne le taux de mots désambiguïsés correctement :

fichier d'apprentissage Brown, 17268 cas ambigus, 14 cat.:	88.6%
fichier de test Brown, 2201 cas ambigus, 14 cat.:	86.3%

4.2 L'aspect langue seconde

Avant de songer à améliorer cette première approche, nous nous sommes posé la question de savoir comment allait s'effectuer le passage du désambiguïsateur du mode monolingue au mode bilingue. Nous avons envisagé deux solutions :

Solution 1 : adapter un DELENE monolingue au corpus L2 par un module supplémentaire. Avantage : contourner la taille restreinte du corpus L2 (27'000 mots). Inconvénient : l'adaptation au mode bilingue.

Solution 2 : apprentissage de DELENE directement sur le corpus L2. Cette solution, plus élégante que la première, est cependant basée sur deux hypothèses : a) la syntaxe particulière utilisée par les apprenants peut être apprise par le perceptron de DELENE et b) les erreurs lexicales et syntaxiques dans les textes n'interfèrent pas de façon majeure avec le processus de désambiguïsation. Avantage : apprentissage directement en mode langue seconde. Inconvénient : la taille réduite de notre corpus L2.

Avant de poursuivre l'une ou l'autre de ces solutions, nous avons fait quelques tests pour essayer de cerner les difficultés rencontrées en passant du mode monolingue au mode "langue seconde".

Test 1 : Pour nous faire une idée de la dégradation que subit DELENE en passant de textes monolingues (apprentissage) à des textes en langue seconde, nous avons évalué la première version de DELENE sur un corpus de test extrait du corpus L2 (il s'agit du corpus de test que nous allons réutiliser jusqu'à la fin de cet article) :

fichier de test L2, 1258 cas ambigus, 14 cat.:	81.2%

La dégradation obtenue, comparée aux 86.3% sur le fichier de test Brown, est d'environ 5% et apparaît bien moins importante que ce que l'on pouvait attendre. En fait, en inspectant les erreurs faites par DELENE, nous avons constaté que seul un pourcentage restreint de ces cas étaient causés par des particularités de la langue seconde.

Test 2 : Nous avons entraîné DELENE directement sur le corpus L2. Pour pouvoir comparer les résultats entre les corpus Brown et L2, nous avons pris des extraits de taille comparable. Voici les résultats :

fichier d'apprentissage Brown, 6134 cas ambigus, 14 cat.:	89.7%
fichier de test Brown, 1403 cas ambigus, 14 cat.:	85.7%
fichier d'apprentissage L2, 5976 cas ambigus, 14 cat.:	93.5%
fichier de test L2, 1258 cas ambigus, 14 cat.:	89.7%

La comparaison indique clairement qu'un DELENE bilingue donne des résultats comparables à un DELENE monolingue pour autant que l'apprentissage se fasse sur un corpus approprié. Les résultats plus élevés sur le corpus L2 peuvent même étonner. Ils sont très certainement dus

aux types de textes qui composent ce corpus ; il faut citer notamment des traductions qui ont pour effet de dupliquer certaines constructions de phrases et une variété de styles moins diversifiée chez les apprenants qui se reposent sans doute surtout sur des structures syntaxiques qu'ils connaissent.

Les résultats obtenus lors de ces deux tests nous ont finalement poussés à opter pour la solution 2 et c'est celle-ci que nous allons utiliser dans le reste de cet article.

4.3 Le nombre de catégories syntaxiques

Pour améliorer les performances de DELENE, nous avons augmenté le nombre de catégories utilisées dans le processus de désambiguïsation (voir figure 2). Voici les catégories de base, puis celles que nous avons ajoutées :

Les 14 catégories de base : Det, Adj, Adv, Nom, V, Prep, Conj, Pron, Gen (les formes du génitif, ex.: a hard day*'s* work), **Int** (l'interjection), **Num** (les numéraux), **NumO** (les terminaisons des cardinaux, ex. 10*th*), **Pct** (les signes de ponctuation), **Del** (le délimiteur de phrase) ;

Les catégories ajoutées : AdvP (adverbe postposé comme dans : to take *off*), **Be** (les formes fléchies de *to be*), **Have** (les formes fléchies de *to have*), **Mod** (les modaux), **Papa** (les participes passés des verbes restants), **DetPron** (pronoms déterminants), **PersPron** (pronoms personnels), **Wh** (pronoms *wh*), **To** (le *to*-infinitif).

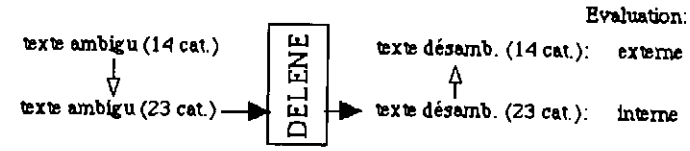

Fig. 2 : augmentation du nombre de catégories : deux types d'évaluation.

Le double but recherché lors de l'affinement des catégories est d'une part de mieux caractériser des séquences de catégories et d'ajouter des contraintes syntaxiques supplémentaires sur le contexte des homographes (ex.: Be, Have, Mod, Papa pour les formes verbales), et d'autre part de

mieux différencier les comportements parfois très variés d'une catégorie en délimitant un comportement précis par une sous-catégorie. Ainsi, la désambiguïsation d'homographes tel que Prep/Adv ou Nom/V peut mieux être apprise en les décrivant comme Prep/AdvP et Nom/Papa. Enfin, à l'aide de cet affinement, nous avons mieux maîtrisé certaines particularités de notre lexique qui parfois gênaient le processus de désambiguïsation. En introduisant, par exemple, la sous-catégorie DetPron, nous avons pu isoler les pré-determinants qui avaient été classés dans le lexique comme pronoms.

Chaque catégorie ajoutée a contribué à améliorer sensiblement les résultats, cependant que le réseau neuronal devenait de plus en plus important et tournait de plus en plus lentement sur notre Mac IIci. Le processus d'affinement pourrait donc être poursuivi, en sous-catégorisant par exemple des catégories qui posent encore des problèmes, mais nous nous sommes arrêtés à un stade où les améliorations devenaient de moins en moins perceptibles et où les résultats en apprentissage se sont très nettement approchés du seuil maximal accessible par la syntaxe locale.

Avant de présenter ces résultats, précisons que nous avons fait la distinction entre une évaluation interne (23 cat.) et externe (les 14 catégories de base), afin de pouvoir faire la comparaison avec les résultats précédents. L'évaluation interne est calculée d'après un plus grand nombre de cas ambigus (ex.: *that* n'est pas ambigu pour l'évaluation externe), dont certains sont faciles à apprendre et augmentent artificiellement les résultats.

Evaluation interne :

- fichier d'apprentissage L2, 7201 cas ambigus, 23 cat.:	96.6%
- fichier de test L2, 1599 cas ambigus, 23 cat.:	92.2%

Evaluation externe :

- fichier d'apprentissage L2, 5818 cas ambigus, 14 cat.:	96.3%
- fichier de test L2, 1238 cas ambigus, 14 cat.:	92.2%

4.4. La durée de l'apprentissage

Parmi les nombreux paramètres qui ont une influence sur le fonctionnement d'un perceptron (citons notamment l'architecture et la topologie du perceptron, différentes constantes d'apprentissage, la taille des corpus d'apprentissage et de test, l'ordre de présentation des situations à apprendre, le hasard, etc.), la durée de l'apprentissage joue un rôle important.

La fin de l'apprentissage ne peut généralement pas être déterminée par un critère précis et cela se passe plutôt intuitivement : après un certain nombre d'essais au cours desquels on a fait varier les paramètres déjà cités, on garde le réseau qui a donné le meilleur résultat. Cependant, si on laisse le perceptron apprendre jusqu'à ce qu'il donne le meilleur résultat sur le corpus d'apprentissage, nous ne pouvons garantir que la généralisation à des textes nouveaux soit optimale. En fait, si on ne l'arrête pas avant, le réseau va, dans la mesure du possible, apprendre les spécificités du fichier d'apprentissage et sa capacité à généraliser va décroître. Pour remédier à cela, nous avons évalué, tout au long de l'apprentissage, un corpus de test et nous avons gardé la version du réseau qui donnait les meilleurs résultats sur celui-ci. Ce critère laisse, bien entendu, deux questions ouvertes : quel texte de test choisir et de quelle taille?

Si on cherche à optimiser le réseau sur le corpus d'apprentissage, on peut par contre se faire une idée des performances du modèle de DELENE, c'est-à-dire de la contribution de la syntaxe locale pure au processus de désambiguïsation. Pour cela, nous avons étendu la durée de l'apprentissage au-delà du meilleur score obtenu à la section précédente :

Evaluation interne et externe :

- fichier d'apprentissage L2 :	98.7%
- fichier de test L2 :	92.0%

Les 98.7% suggèrent que le perceptron est parfaitement capable de réaliser le modèle de la syntaxe locale, car il est difficile de s'imaginer que ce modèle puisse être encore plus performant. La généralisation quant à elle n'a pas réussi à suivre le même développement et s'est quelque peu dégradée, comme on doit s'y attendre.

5. Apprentissage de connaissances morphologiques

Dans ce qui précède, nous sommes partis du fait que nous avions à traiter des mots ambigus dont les catégories sont connues de notre lexique. La réalité veut cependant que l'on soit sporadiquement confronté à des mots inconnus (soit parce que tout lexique est limité, soit parce que certains mots mal orthographiés n'ont pas pu être corrigés par un correcteur orthographique). La solution classique procède à une analyse morphologique du mot pour proposer une ou plusieurs catégories au processus de désambiguïsation. Pour notre réseau, une autre solution consisterait, par exemple, à activer toutes les catégories dans CATS(i) et à choisir parmi celles-ci la plus activée en sortie. Un test préliminaire que nous avons effectué sur cette base et sur les mêmes textes a livré des résultats entre 50% et 60%, ce qui est très nettement insuffisant.

Pour notre approche, nous avons voulu savoir si un perceptron pouvait utiliser des informations morphologiques et les combiner avec des informations syntaxiques. Pour cela, nous avons échangé le bloc CATS(i) utilisé jusqu'à présent par un bloc TERM(i) contenant la terminaison du mot d'une longueur de quatre lettres (voir figure 3 ci-dessous). Chaque lettre est codée sur quatre neurones et le codage ajusté par le réseau pendant l'apprentissage (pour plus de détails, voir Miikkulainen & Dyer (1991) ; Bodmer (1992)).

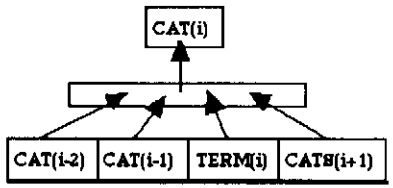

Fig. 3 : version avec bloc de terminaison

Nous nous sommes attendus à ce que le réseau parvienne à coder les données alphabétiques dans le bloc TERM et qu'il apprenne à extraire, parmi les terminaisons de longueur fixe, les séquences de lettres pouvant l'aider. Voici, sur les mêmes textes qu'auparavant, les résultats obtenus avec ce modèle :

fichier d'apprentissage L2, 7201 cas ambigus, 23 cat.:	88.6%
fichier de test L2, 1599 cas ambigus, 23 cat.:	82.7%

Ce résultat est satisfaisant surtout si l'on considère que seules quelques formes fléchies dans un nombre restreint de catégories peuvent être

reconnues à partir de leur suffixe. Pour vérifier que ce résultat n'était pas dû en majeure partie au contexte syntaxique, nous avons entraîné et testé le même réseau en maintenant TERM(i) à zéro. Les résultats obtenus sur les mêmes textes étaient nettement moins bons :

fichier d'apprentissage L2, 7201 cas ambigus, 23 cat.:	53.2%
fichier de test L2, 1599 cas ambigus, 23 cat.:	52,7%

Ces chiffres soulignent l'importance du bloc TERM(i) dans son rôle qui est de proposer des catégories syntaxiques à la place du bloc CATS(i). Nous n'avons pas la place ici pour discuter la nature des connaissances morphologiques apprises par le réseau. Le lecteur intéressé trouvera dans Elenius (1989) une étude comparable sur le suédois.

6. Conclusion

L'objectif de cet article était de présenter un désambiguïsateur lexical neuronal. Nous nous sommes principalement penchés sur les résultats suivants : le nombre de catégories à utiliser, la désambiguïsation de texte en langue seconde et l'apprentissage de connaissances morphologiques.

La première version du désambiguïsateur est un modèle neuronal utilisant uniquement la syntaxe locale sur une fenêtre de quatre mots. Nous avons pu vérifier qu'en apprenant à désambiguïser plus de 98% des cas d'ambiguïté de notre corpus L2, ce modèle rendait bien compte de l'aspect décisif de la syntaxe locale dans ce processus. Cependant, pour pouvoir utiliser notre désambiguïsateur, nous avons retenu une version qui n'était pas optimisée sur le corpus d'apprentissage, mais sur un corpus de test. Les résultats, sur le corpus d'apprentissage et de test, sont respectivement de 93.5% et 89.7% en utilisant les 14 catégories de base et de 96.3% et 92.2% en utilisant 23 catégories. Cette amélioration s'explique par le fait que la sous-catégorisation des catégories de base mène à un système où chaque catégorie tend à n'exprimer qu'un seul comportement distinct, ce qui facilite l'apprentissage du réseau. L'obtention d'un corpus de taille plus grande serait un atout majeur pour améliorer encore ces résultats.

Ensuite, nous avons constaté que notre désambiguïsateur pouvait être utilisé tout aussi bien sur des textes d'apprenants que sur des textes de

locuteurs natifs, pour autant que l'apprentissage se fasse sur des textes correspondants. La syntaxe particulière de ce corpus et les nombreuses erreurs lexicales et syntaxiques n'ont pas interféré de façon significative avec le processus de désambiguïsation.

Enfin, nous avons créé une version *hybride* de notre désambiguïsateur en remplaçant l'information catégorielle du mot ambigu par sa terminaison. Les résultats ainsi obtenus, soit 88,6% et 82.7%, soulignent qu'en grande partie, le réseau peut apprendre à extraire de la terminaison d'un mot l'information reliée à ses catégories syntaxiques et à mélanger cette information avec les connaissances d'un autre niveau.

7. Bibliographie

BENELLO, J., A.W. MACKIE & J.A. ANDERSON (1989): "Syntactic category disambiguation with neural networks", *Computer Speech and Language*, 3, 203-217.

BODMER, F. (1992): *DELENE, un désambiguïsateur lexical neuronal pour micro-ordinateur*, Mémoire pour l'obtention du diplôme postgrade en informatique technique, Lausanne, EPFL.

BRILL, E. (1992): "A simple rule-based part of speech tagger", *Proceedings of the 3rd Conference on Applied NLP (ACL)*, Trento, 152-155.

CHURCH, K.W. (1988): "A stochastic parts program and noun phrase parser of unrestricted text", *Proceedings of the 2nd ANLP Conference*.

CRUCIANU, M. & D. MEMMI (1992): "Extracting the implicit structure in a connectionist network", *NeuroNîmes 1992*, 491-502.

CUTTING, D., J. KUPIEC, J. PEDERSEN & P. SIBUN (1992): "A practical part-of-speech tagger", *Proceedings of 3rd Conference on Applied NLP*, Trento, 133-140.

DEROSE, S.J. (1988): "Grammatical category disambiguation by statistical optimization", *American Journal of Computational Linguistics*, 14 (1), 31-39.

DERTHICK, M. (1990): "Mundane reasoning by settling on a plausible model", in: HINTON, G. (Ed.), *Connectionist Symbol Processing*, MIT/Elsevier.

EIZIRIK, M.R., V.C. BARBOSA & S.B.T. MENDES (1993): "A Bayesian-network approach to lexical disambiguation", *Cognitive Science*, 17.

ELENIUS, K. & R. CARLSON (1989): "Assigning parts-of-speech to words from their orthography using a connectionist model", *Proceedings of the European Conference on Speech Communication and Technology*, 1, 534-537.

ELMAN, J.L. (1991): "Distributed representations, simple recurrent networks, and grammatical structure", *Machine Learning*, 7, Boston, Kluwer Academic Publishers.

ELMAN, J.L. (1989): "Finding structure in time", *Cognitive Science*, 14, 179-211.

ELMAN, J.L. (1989): "Representation and structure in connectionist models", *CRL Technical Report 8903*, Center for Research in Language, University of California, San Diego.

FAISAL, K.A. & S.C. KWASNY (1990): "Design of a hybrid deterministic parser", *COLING*, 11-16.

FANTY, M. (1986): "Context-free parsing with connectionist networks", in: DENKER, J.S., *AIP Conference Proceedings N° 151*, New York, American Institute of Physics.

GARSIDE, R., G. LEECH & G. SAMPSON (Eds.) (1987): *The Computational Analysis of English: A Corpus Base Approach*, London, Longman.

GASSER, M. & M.G. DYER (1988): "Sequencing in a connectionist model of language processing", *COLING*, 185-190.

GREENE, B.B. & G.R. RUBIN (1971): "Automatic grammatical tagging of English", Unpublished manuscript, Providence, Rhode Island, Department of Linguistics, Brown University.

HERTZ, J., A. KROGH & R.G. PALMER (1991): *Introduction to the Theory of Neural Computation*, Redwood, Addison-Wesley.

HINTON, G. (1990): "Mapping part-whole hierarchies into connectionist networks", in: HINTON, G. (Ed.) *Connectionist Symbol Processing*, Amsterdam, MIT/Elsevier.

JAIN, A.N. (1991): "Parsing complex sentences with structured connectionist networks", *Neural Computation*, 3 (1), 110-120.

KLEIN, S. & R.F. SIMMONS (1963): "A computational approach to grammatical coding of English words", *Journal of the Association for Computing Machinery*, 10.

MARTIN, W., R. HEYMANS & F. PLATTEAU (1988): "DILEMMA, an automatic lemmatizer", in: MARTIN, W. (Ed.) *COLINGUA 1, Antwerp Papers in Linguistics*,56, Antwerp, 5-62.

MIIKKULAINEN, R. & M.G. DYER (1991): "Natural language processing with modular PDP networks and distributed lexicon", *Cognitive Science*, 15, 343-399.

NAKAGAWA, H. & M. TATSUNORI (1988): "A parser based on connectionnist model", *COLING*, 454-458.

PAULUSSEN, H. (1992): *Automatic Grammatical Tagging: Description, Comparison and Proposal for Augmentation*, Licentiaat in de Linguïstiek door, Universiteit Antwerpen.

POLLACK, J.B. (1990): "Recursive distributed representation", in: HINTON, G. (Ed.), *Connectionist Symbol Processing*, Amsterdam, MIT/Elsevier.

RAGER, J. & G. BERG (1990): "A connectionist model of motion and government in Chomsky's government-binding theory", *Connection Science*, 2.

RUMELHART, D.E., G.E. HINTON & R.J. WILLIAMS (1986): "Learning representations by back-propagation errors", *Nature*, 323.

SCHOLTES, J. (1991): "Learning simple semantics by self-organisation", in: POWERS, D. & L. REEKER (Eds.), *Machine Learning of Natural Language and Ontology*, Deutsches Forschungs-zentrum für KI.

SMOLENSKY, P. (1990): "Tensor product variable binding and the representation of symbolic structures in connectionist systems", in: HINTON, G. (Ed.), *Connectionist Symbol Processing*, Amsterdam, MIT/Elsevier.

ST.JOHN, M.F. & J.L. McCLELLAND (1990): "Learning and applying contextual constraints in sentence comprehension", in: HINTON, G. (Ed.), *Connectionist Symbol Processing*, Amsterdam, MIT/Elsevier.

TOURETZKY, D.S. (1990): "BoltzCONS: dynamic symbol structures in a connectionist network", in: HINTON, G. (Ed.), *Connectionist Symbol Processing*, Amsterdam, MIT/Elsevier.

VAN GELDER, T. (1989): *Distributed Representation.*, Unpublished doctoral dissertation, Departement of Philosophy, University of Pittsburgh.

VERONIS, J. & N.M. IDE (1990): "Word sense disambiguation with very large neural networks extracted from machine readable dictionaries", *COLING*, 389-394.