RECONNAISSANCES PLURI-LEXICALES DANS CELINE, UN SYSTÈME MULTI-AGENTS DE DÉTECTION ET CORRECTION DES ERREURS

Jacques MENÉZO
Damien GENTHIAL
Jacques COURTIN

Équipe TRILAN
Laboratoire CLIPS
Institut de Mathématiques Appliquées de Grenoble

BP 53, F-38041 Grenoble Cedex 9, FRANCE

Jacques.Menezo@imag.fr

RÉSUMÉ :

Cet article se propose de présenter la reconnaissance lexicale dans C.E.L.I.N.E., système automatique ou interactif de détection correction des erreurs lexicales et syntaxiques, basé sur une architecture se prévalant de l'I.A.D. et des systèmes multi-agents. Une gestion dynamique des accointances des agents, la prise en compte de leurs expériences, permet à tout instant d'ajuster les stratégies aux agents disponibles, à l'utilisateur, au type de texte traité (type autodéfini par le système) et même à la fenêtre active sur le texte

MOTS CLEFS :

Correction automatique ; Erreurs lexicales ; Erreurs syntaxiques ; Multi-Agents ; Intelligence Artificielle Distribuée ; Architecture Répartie

1. INTRODUCTION

La détection correction d'erreurs présente plusieurs phases (lexicale, syntaxique, sémantique, pragmatique, phonétique...) et demande pour être optimisée ou même simplement efficace, non pas un ordonnancement séquentiel de ces phases mais un traitement en parallèle. Les langues naturelles présentent un haut degré d'ambiguïté, degré encore augmenté en présence d'erreurs. Ces ambiguïtés interdisent pratiquement qu'une seule des phases de l'analyse aboutisse au niveau des traits manipulés par cette phase à une solution unique désambiguïsée. Une vérification-correction complète va demander une collaboration de l'ensemble des traits pertinents disponibles, à travers les différentes phases, pour éliminer le maximum de solutions concurrentes. Pour mettre en œuvre cette complémentarité, on peut envisager un système réparti d'agents spécialisés possédant chacun ses connaissances et collaborant à cette tâche (Stéphanini 93; Letellier 93, Genthial 94). Dans la terminologie des architectures multi-agents et de l'I.A.D., les agents sont organisés en société, et, de ce fait, par ses connaissances et ses relations, chaque agent possède un comportement social, comportement qui tend dans certaines réalisations de l'I.A.D. à reproduire certains aspects du comportement humain. Les connaissances et le comportement de l'agent peuvent être évolutifs. Chaque agent peut posséder un aspect cognitif plus ou moins marqué ou au contraire être purement réactif (Demazeau 92; Occello 93). L'intelligence du système émerge de l'activité globale résultant de la collaboration entre agents et provient aussi de l'évolution des connaissances et de l'expérience de chaque agent. Il va falloir déterminer des tables de correspondance entre les traits manipulés par un agent « extérieur » et les traits en vigueur dans CELINE. Dans l'hypothèse de nombreux agents, il n'est pas indispensable que la correspondance des traits soit totale ou très rigoureuse, la complémentarité entre agents fait qu'un renseignement même fragmentaire amené par un agent peut être utile pour la levée des ambiguïtés.

Nous présentons d'abord succinctement l'architecture générale du système puis nous décrivons plus en détails la reconnaissance lexicale avant d'aborder le problème précis du traitement des mots inconnus.

2. ARCHITECTURE DE CELINE

Sur le plan du contrôle, l'architecture (figure 1) est une architecture hiérarchique et pyramidale :

- Un *superviseur* coordonne l'action de plusieurs *pilotes.*
- Chaque pilote est responsable d'un *secteur d'activité* (lexicale, syntaxique, accords en nombre et en genre, suivi statistique, etc.) et coordonne le travail de plusieurs *agents de travail*.
- Chaque agent est spécialiste d'un domaine étroit mais peut dépendre de plusieurs pilotes.
- L'agent humain communique avec le superviseur, les pilotes et des agents à caractères cognitifs.
- Certains agents travaillent d'une façon indépendante à la demande (non représenté à la figure 1). On peut envisager un agent autonome spécialisé dans une tâche comme par exemple le traitement des bulletins météo incluant un traitement sémantique et fournissant « clef en main » la version corrigée de la phrase ou du paragraphe.

EXEMPLE DE COLLABORATIONS ENTRE AGENTS

La correction d'une phrase d'un élève de quatrième (rédaction avec utilisation d'un traitement de texte) « *le baeu chevax noir racée et les juents courrrent furieux et*

affamée ») permet d'illustrer la collaboration entre agents

•Pour la correction de *baeu* un pilote des activités lexicales PAL interroge un agent d'analyse morphologique PILAF (Courtin 92) qui lui fournit deux solutions *beau* (adjectif ou substantif) et *bau* (substantif, largeur d'un bateau).

•De la même façon, pour la correction de *chevax,* le pilote PAL et l'agent PILAF trouvent une solution *chevaux*.

fournit les graphies correctes des mots corrigés.

•Optionnel : un agent FORME formate le texte avec un seul blanc entre deux mots consécutifs et ajoute une majuscule en début de phrase.

Au total la phrase est correctement corrigée en « *Le beau cheval noir racé et les juments courent furieux et affamés* ».

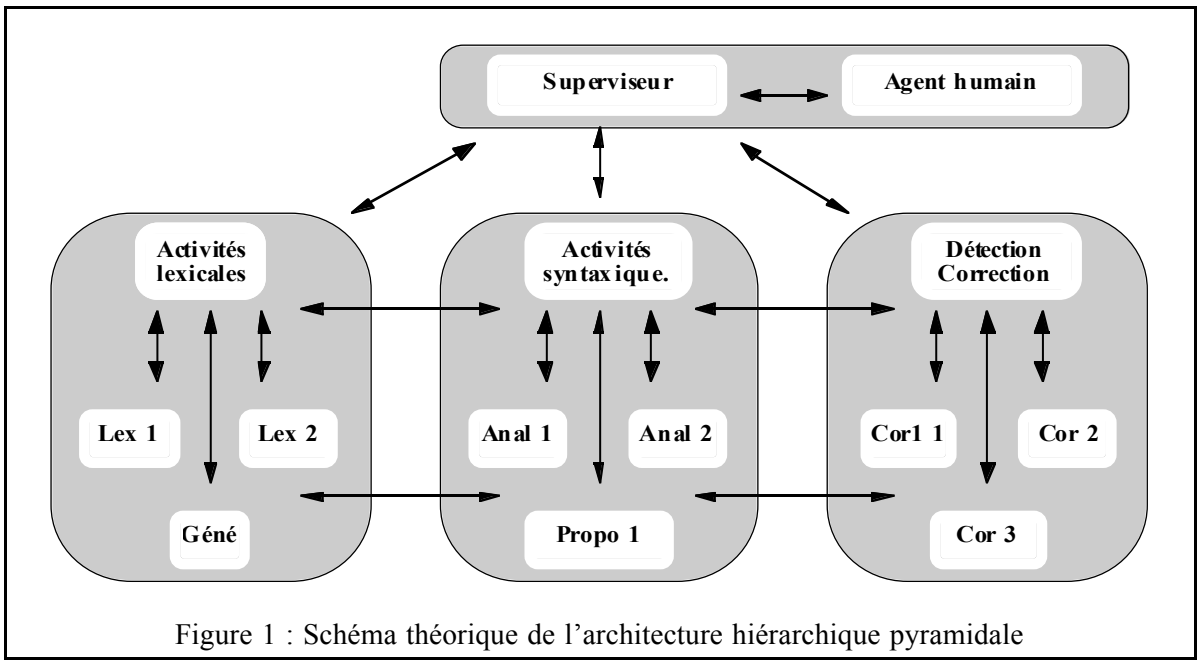

Figure 1 : Schéma théorique de l'architecture hiérarchique pyramidale

•Un pilote des activités syntaxiques (PAAS) et des agents d'analyse syntaxique valident la séquence *le beau chevaux (déterminant adjectif substantif)* et refusent *le bau chevaux (déterminant substantif substantif).*

•Un agent (PSCM, proposition statistique de catégories morphologiques) utilisant les modèles cachés de Markov et une matrice de transition par défaut, peut aider au choix entre *bau* et *beau* (utilisation d'un modèle triclasse avec probabilité élevée des transitions *det adjq subc* au dessus du seuil de confiance et probabilité quasi nulle de la transition det subc subc) (Kallas 87, Menézo 92).

•Un pilote (PAVA) de l'activité de vérification des accords et un agent (DCFA) de détection et correction des fautes d'accords corrigent les accords en utilisant une méthode particulière dite méthode des structures (Menézo 96).

•Le générateur morphologique de PILAF

QUELQUES CARACTÉRISTIQUES DE CELINE

CELINE est un système ouvert présentant quelques originalités :

1. Dans le secteur lexical, la possibilité d'un accès optimisé à de très nombreux agents de reconnaissances lexicales implantés éventuellement sur des sites lointains (pour fixer les idées disons plusieurs dizaines). Cette possibilité a été recherchée compte-tenu de la diversité et de la spécificité des agents lexicaux disponibles (par exemple (Maurel 89), automates sur la reconnaissance des adverbes de temps, de dates, de noms de villes ...).

2. Dans le secteur syntaxique également, la possibilité d'admettre de nombreux analyseurs travaillant avec des grammaires différentes. Un ensemble de ***structures*** liées par une relation d'ordre (par exemple sous forme d'ordre lexicographique des chemins) permet de disposer

d'une relation interne commune.

3. Dans le secteur de la détection-correction des fautes d'accords, une méthode quantifiée basée sur les structures.

4. L'heuristique mise en jeu dans CELINE repose généralement sur des connaissances acquises par le biais de statistiques et quantifiées par un ensemble de coefficients. Cette approche s'inscrit dans un changement de cap de la linguistique informatique vers les approches statistiques et probabilistes ainsi que pour celles qui marient symbolique et quantitatif (Atala 95).

3. RECONNAISSANCE LEXICALE

3.1 LES AGENTS LEXICAUX

Le pilote PAL va distribuer la tâche d'identifier chacun des mots (et de fournir racines, catégories et variables morphologiques) aux différents agents du secteur d'activités (que nous appellerons *agents lexicaux*). Ce pilote pourra également intervenir lors d'une étape de génération après une correction. Ces agents peuvent être éligibles ou pas selon le paramétrage du système, paramétrage défini par l'utilisateur selon le(s) domaine(s) du texte traité et le(s) domaine(s) de leurs bases de connaissances.

N° agent	*Nom*	*Domaine*
1	PILAF	Vocabulaire courant
2	NOPRO	Noms propres
3	CHIMOR	Chimie Organique
4	ELEC	Électricité
5	DICPERS	Dictionnaire(s) personnel(s)
6	DICKPERSN	Noms propres
7	TJRP	Fautes habituelles

Figure 2: Exemples de lexiques

Pour certains domaines très spécialisés, le pilote PAL pourra s'adresser à un agent monolithique indépendant assurant l'ensemble des traitements (reconnaissance lexicale, analyse syntaxique, analyse sémantique) et fournissant une réponse finale sur la phrase ou le paragraphe envisagés.

3.2 COEFFICIENTS D'UTILISATION PAR SESSION

3.2.1 GÉNÉRALITÉS

Le **coefficient d'utilisation de l'agent lexique n°i** représente la probabilité à priori pour qu'un mot quelconque soit reconnu par le lexique n°i. Ce coefficient permettra de déterminer l'ordre d'utilisation des lexiques pour la correction d'un mot donné.

Nous appellerons **session** le traitement d'une partie plus ou moins longue de texte(s) et nous désignerons une session par une lettre F, T, U, G.

F	**F**enêtre sur le texte. Cette fenêtre constitue un échantillon de l'ordre de la centaine de mots soit environ un tiers de page à une demi page.
T	**T**exte entier. Par exemple, un chapitre de thèse, un article, une lettre.
U	**U**tilisateur La session est l'ensemble des textes de l'utilisateur.
G	**G**énéral Il s'agit cette fois de l'ensemble des textes examinés par le système.

Figure 3 : Liste des sessions

Selon la session envisagée, les **C**oefficients d'**U**tilisation vont être nommés respectivement CU**F**, CU**T**, CU**U**, CU**G**. Nous utiliserons ces abréviations par la suite.

Pour illustrer l'aspect dynamique (en caricaturant « en temps réel ») du suivi du texte, nous garderons en mémoire l'exemple d'un chercheur spécialiste de chimie organique travaillant dans le domaine de l'agriculture et écrivant un article comportant dans l'ordre une introduction, un historique, un développement technique (avec inclusion d'expressions météorologique), une conclusion et pour terminer une bibliographie.

3.2.2 COEFFICIENTS D'UTILISATION PAR FENÊTRE

Nous distinguerons deux cas : le traitement initial d'un nouveau texte et la reprise d'un texte :

Texte nouveau

Les coefficients vont être obtenus à partir d'un échantillon du texte (la fenêtre). Diverses stratégies sont utilisables :

1. La plus souple (inévitable pour un système sans expérience) : les CUF sont tous initialisés à zéro (refus d'idées préconçues sur le type de texte) tous les X mots (pour fixer les idées X de l'ordre de la centaine). Dans l'exemple, si on

passe sur la partie historique ou sur la bibliographie, le coefficient va montrer une utilisation élevée du lexique des noms propres alors que sur le développement technique, le coefficient montrera une utilisation fréquente du lexique de la chimie organique ou de celui de la météorologie.

2. La plus efficace en moyenne pour un système avec expérience : initialiser les CUF avec des valeurs moyennes (cf. coefficient d'utilisation par utilisateur CUU et à défaut coefficient d'utilisation général CUG).

Reprise de texte et feuilles lexico-syntaxiques

Les CUF entraîne un « typage » local du texte. On peut envisager (tous les X mots ou à chaque changement de titre pour un texte avec plan) de mémoriser la valeur des coefficients. Lors d'une reprise du texte, la récupération de ces coefficients va permettre d'obtenir une certaine optimisation immédiate sans attendre une convergence à partir de valeurs arbitraires.

Dans le cas du chercheur en chimie, il est probable que tous les paragraphes d'un même niveau de différents articles vont montrer des utilisations des lexiques comparables (par exemple pour l'introduction, pour la conclusion, pour la bibliographie).

La mémorisation des CUF pourrait s'intégrer dans ***des feuilles lexico-syntaxiques*** comparables en vue de la détection-correction des erreurs aux feuilles de styles pour la mise en forme d'un texte. Ce typage du texte sera aussi utilisé au niveau syntaxique.

En descendant vers une granularité plus fine, il est possible de mémoriser le lexique optimum pour chaque mot. Dans le cadre d'un logiciel moderne, cela ne semble pas plus exigeant que la possibilité de mémoriser les marques de formats du style (gras; italique, souligné, police, taille, ...) pour chaque caractère.

3.2.3 COEFFICIENTS D'UTILISATION PAR TEXTE

Les principes de calcul restent les mêmes mais cette fois le cumul des reconnaissances par lexique est fait au niveau du texte entier.

La mémorisation de ces coefficients va permettre lors d'une **reprise du même texte** d'initialiser les CUF à des valeurs non arbitraires et de converger plus vite vers des valeurs efficaces. Pour comprendre l'utilité de cette possibilité il faut se rappeler le cadre du système avec de très nombreux agents lexicaux accessibles.

Le coefficient d'utilisation par texte représente pour un lexique la moyenne des coefficients par fenêtre de ce lexique pour ce texte. Ces coefficients représentent aussi une forme d'acquisition d'une expérience par le système.

3.2.4 COEFFICIENTS D'UTILISATION PAR UTILISATEUR

La mémorisation de ces coefficients va permettre lors du traitement d'**un nouveau texte** d'initialiser les CUF à des valeurs non arbitraires en permettant en moyenne une optimisation.

Le coefficient d'utilisation par utilisateur représente pour un lexique la moyenne des coefficients par textes de ce lexique pour cet utilisateur.

3.2.5 COEFFICIENTS D'UTILISATION GÉNÉRAUX MOYENS

Il va permettre une initialisation optimisée cette fois pour un **nouvel utilisateur**. Par exemple, imaginons un système de détection-correction utilisé par les chercheurs d'une même équipe. Le système a acquis une expérience et une connaissance de ces chercheurs et va donc «réagir intelligemment» face à un nouvel utilisateur.

Le coefficient d'utilisation général moyen représente pour un lexique la moyenne des coefficients par utilisateur pour ce lexique.

3.3 LES COEFFICIENTS D'EFFICACITÉ PAR SESSION

Nous nous plaçons cette fois à un autre point de vue que celui de la simple reconnaissance par un lexique : celui de la pertinence des traits délivrés en vue de l'analyse syntaxique et de la détection correction des fautes d'accords. Cette évaluation est importante puisque le système se propose d'intégrer des agents lointains éventuellement imparfaits et dont les traits sont plus ou moins adaptés à ceux utilisés par CELINE. Il s'agit donc de quantifier par des coefficients les lexiques qui procurent un ensemble de traits permettant l'application de la méthode des structures.

Le mode de calcul des coefficients d'efficacité (CE) est analogue à celui présenté pour le

coefficient d'utilisation. Nous ne développerons pas les variantes selon les différentes sessions (fenêtres, texte, utilisateur, général), l'interprétation est à peu près équivalente à celles des coefficients d'utilisation.

3.4 LES COEFFICIENTS DE CRÉDIBILITÉ PAR FENÊTRE

Définition du coefficient de crédibilité

Ce coefficient va exprimer l'intérêt du système à utiliser les n lexiques i,j,k, ... en priorité; l'utilisation de ces lexiques permettant d'optimiser tout à la fois la reconnaissance lexicale et la pertinence des informations en vue de la détection-correction des erreurs par la méthode des structures.

Calcul du coefficient de crédibilité (CC) :

Pour une session X avec X ∈ {F, T, U, G} :

CCX = CUX * CEX

Utilisation du coefficient de crédibilité

Ce coefficient semble surtout intéressant en mode interactif pour diminuer le temps de réponse du système en évitant pour l'utilisateur des temps d'attente trop long. De ce fait, la session la plus intéressante semble être la fenêtre.

4. TRAITEMENT D'UN MOT INCONNU

Face à un mot inconnu (donc rejeté par tous les lexiques disponibles) le pilote PAL va disposer d'un certain nombre de voies de recherche :

1) La génération morphologique

L'analyse morphologique initiale du mot inconnu, bien qu'ayant échouée, peut avoir fourni des indices tels que racine(s) possible(s) et des variables morphologiques (genre, nombre, etc.).

Différents agents (PSCM par la modélisation de Markov; PAS par l'analyse syntaxique) peuvent proposer de leurs cotés une ou plusieurs catégories morphologiques. Le module de vérification des accords peut fournir, en s'appuyant sur les structures reconnues, des renseignements sur les variables morphologiques voir même sur les catégories morphologiques. Au total, si on dispose de « racine + catégorie morphologique + variables morphologiques » on peut alors demander au générateur morphologique de proposer une solution.

Faute supposée	Traitement
Lettre triple	On supprime une des trois lettres et en cas d'un nouvel échec de l'analyse on en supprime deux. (Exemple : têtttard pour têtard)
Lettre double	On en supprime une (Exemple : propposition pour proposition)
Permutation de deux lettres	On permute les lettres adjacentes deux à deux (Exemple : volie pour voile; on va essayer successivement : ovlie, vloie, voile)
Lettre manquante	On essaye successivement les 26 lettres de l'alphabet à toutes les positions possibles. (Exemple : accient pour accident, on va essayer Xaccient avec X égal successivement à a,b,c,d, ...x,y,z puis aXccient, acXcient, accXient, acciXent)
Substitution de lettres	On remplace successivement chaque lettre par les 26 lettres.

Figure 4: Traitements alphabétiques

2) Outre la génération morphologique, nous disposons d'un **agent de reconnaissance par clef squelette** et d'un **agent de reconnaissance par clef phonétique** (Strube de Lima 90).

3) Traitement alphabétique direct : suppression, ajout, permutation de lettres (figure 4)

A l'initialisation du système, le paramétrage permet de faire le choix des méthodes retenues ainsi que de l'ordonnancement de leurs applications. Ensuite par le biais de la statistique, le système va évoluer librement et l'ordonnancement tiendra compte de l'efficacité de chaque méthode. Les méthodes mises en

œuvre pour l'instant supposent une seule faute par mot.

Remarques :

1. Les recherches par lettre manquante ou substitution de lettres élèvent terriblement le nombre de solution concurrentes. En attendant des essais plus élargis, une heuristique provisoire est de ne les activer qu'en cas d'échec des autres traitements.

2. Avec certains lexiques (par exemple PILAF), certaines méthodes disponibles comme celle de la clef squelette corrigent automatiquement une partie des fautes du tableau précédent. Nous avons toutefois conservé la possibilité de choisir au niveau du pilote PAL, l'ensemble des traitements ci-dessus de manière à pouvoir les appliquer lors de la rencontre d'un système lexical ne possédant pas de système de correction par lui-même (par exemple, le lexique de chimie organique).

3. Lorsqu'un mot n'est pas reconnu, on peut essayer de le scinder en deux mots en introduisant un espace successivement entre toutes les lettres qui le compose.

4. Lorsque deux mots successifs d'une même phrase ne sont pas reconnus on essaye diverses solutions telles que concaténation des deux mots ou concaténation de : premier mot, une lettre X et le deuxième mot avec X égal successivement à a,b,c,d, ...x,y,z.

Le filtrage des solutions concurrentes est réalisé à partir d'un ensemble de coefficients de pondération ainsi que par le calcul de distances entre solutions concurrentes et graphie initiale.

5. CONCLUSION

Le pilote peut accéder à divers lexiques soit sur la machine hôte soit sur des sites distants. En vue d'une optimisation cette multiplicité de lexiques demande une planification des recherches. A travers des statistiques et un jeu de coefficients, une gestion dynamique des accointances du pilote PAL et la prise en compte de son expérience, permettent à tout instant d'ajuster les stratégies aux agents lexicaux disponibles, à l'utilisateur, au type de texte traité (type auto défini par le système) et même à la fenêtre active sur le texte.

Le système peut fonctionner soit en interaction permanente avec l'agent humain, soit en mode automatique. Le fonctionnement complètement automatique face à un texte nouveau et un utilisateur nouveau faisant de nombreuses fautes de compétence et de performance est bien entendu difficile. Par contre le traitement automatique peut-être envisagé dans un certain nombre de cas (par exemple : pour un utilisateur identifié et connu, voire même pour un utilisateur inconnu sur la machine d'une équipe, ou pour des relectures multiples ou encore pour un domaine restreint tel que le vocabulaire réduit d'une interface homme-machine).

Nous proposons aussi une mémorisation des divers coefficients et de certains repères soit comme caractéristique de chaque mot pour un texte souvent repris soit dans des feuilles de style lexico-syntaxiques.

RÉFÉRENCES

(Atala 95) Revue semestrielle de l'ATALA, *Traitements probabilistes et corpus*, volume double n°36, 1995.

(Cohard 88) B. Cohard, *Logiciel de détection et de correction des erreurs lexicale,* Mémoire d'ingénieur CNAM, Centre régional associé de Grenoble, Mars 88.

(Courtin & Dujardin 92) J. Courtin, D. Dujardin, D. Genthial, I. Kowarski, *Outils lexicaux de l'équipe TRILAN, bilans et perspectives,* Les actes des journées GRECO,PRC, Communication Homme-machine, Séminaire LEXIQUE, Toulouse , Janvier 92.

(Demazeau 92) Y. Demazeau, *Modèle d'Agent Cognitif Hiérarchique (COHIA),* Rapport interne du LIFIA, 1992.

(Genthial & Courtin 94) D. Genthial et J. Courtin, *Towards a More User-Friendly Correction,* 15th CoLing, Kyoto, Japan, August 1994, pp 1083-1088.

(Kallas 87) G. Kallas, *Résolution des solutions multiples en Analyse Morphologique Automatique de Langues Naturelles, Utilisation des Modèles de Markov,* Thèse de doctorat d'état, Centre de Recherche en Informatique Appliquée aux Sciences Sociales, Grenoble, Juin 87.

(Letellier 93) S. Letellier, *ECLAIR, un système d'analyse et de correction lexicales multi-experts et multi-lexique,* Thèse de doctorat, Université de Paris-sud, Centre d'Orsay, Décembre 1993.

(Maurel 89) D. Maurel, *Reconnaissance de*

séquences de mots par automates, adverbe de date du français, Thèse de doctorat en informatique, Université de Paris VII, 1989.

(Menézo 92) J. Menézo, *Désambiguïsation lexicale par filtrages en cascade.* Les actes des journées GRECO,PRC, Communication Homme-machine, Séminaire LEXIQUE, Toulouse, Janvier 92.

(Menézo 96) J. Menézo, *La méthode des structures, principe et mise en œuvre dans CELINE,* TALN'96, Marseille, Mai 96.

(Occello 93) M. Occello, *Blackboards Distribués et Parallèles : Application au Contrôle de Systèmes Dynamiques en Robotique et en Informatique musicale,* Thèse de Doctorat, Université de NICE,SOPHIA ANTIPOLIS, Janvier 93.

(Rajman 95) M. Rajman. *Approche probabiliste de l'analyse syntaxique,* Traitements probabilistes et corpus, Volume 36, Revue semestrielle de l'ATALA, 1995.

(Stéphanini 93) M.-H. Stéphanini. *TALISMAN : une architecture multi-agents pour l'analyse du français écrit.* Thèse de doctorat, Université Pierre Mend*ès*, France, Grenoble, Janvier 1993.

(Strube de lima 90) V. L. Strube de Lima, Contribution à l'étude du traitement des erreurs au niveau lexico-syntaxique dans un texte écrit en français, Thèse de l'Université Joseph Fourrier, Grenoble 1, Mars 1990.