Les cadres théoriques des TAL syntaxiques:

quelle adéquation linguistique et algorithmique ?

une étude et une alternative

Jacques Vergne

Jacques.Vergne@info.unicaen.fr

GREYC - CNRS URA 1526 - Université de Caen F-14032 Caen cedex France

Résumé introductif

Dans l'évolution d'une technique, un certain cadre théorique permet une avancée à un certain moment de son histoire, puis des inadéquations apparaissent; pour les dépasser, il faut les étudier, les reconnaître, puis rechercher des alternatives.

Dans la première partie, on étudie ces inadéquations, d'abord linguistiques (objets et objectifs différents, démarches opposées), puis algorithmiques (pourquoi les algorithmes sont combinatoires).

Dans la deuxième partie, on présente une solution alternative: comment construire une théorie syntaxique de l'objet traité: la part concrète d'une langue; par quels principes algorithmiques on peut éviter toute combinatoire et obtenir des algorithmes de complexité linéaire en temps.

***Étude** des (in)adéquations linguistique et algorithmique des cadres théoriques des TAL syntaxiques*

Deux pratiques des Traitements Automatiques des Langues (TAL)

- traiter l'objet concret pour agir: les TAL comme technique opératoire (*informatique* linguistique)

C'est la pratique la plus courante: le but est opératoire, mais en tentant souvent d'éviter le détour théorique (comme au cours de toute l'histoire du TAL); cette pratique est principalement syntaxique.

- traiter pour comprendre: l'outil de recherche en linguistique (*linguistique* informatique)

On est alors dans le cadre d'une recherche fondamentale en linguistique: expérimenter, observer, modéliser avec l'aide de l'ordinateur. Cette démarche est féconde et prometteuse (autant que l'apparition de la lunette astronomique en astronomie), mais elle est encore rare (par exemple, travaux de Catherine Fuchs et Bernard Victorri sur la modélisation de la polysémie, travaux de Laurent Gosselin sur la sémantique de la temporalité, travaux de Jacques Vergne sur la syntaxe des langues concrètes).

Faut-il un cadre théorique aux TAL?

La nécessité d'un cadre théorique n'est pas admise par toute la communauté:

- démarche scientifique: comprendre pour comprendre, et comprendre pour agir

On fait l'hypothèse que l'action est d'autant plus efficace que la compréhension est fine.

Quel est l'objet traité, et donc à étudier et comprendre? ce sont les textes écrits, les langues sous cet aspect restreint concret, matériel, et non pas le langage: ni la compétence, ni la performance.

- analyse par fréquences statistiques acquises automatiquement sur corpus:

Ces travaux n'ont pour le moment pas de base théorique linguistique, avec un objectif purement opératoire (des bases théoriques statistiques sont utilisées dans des outils statistiques, mais ne théorisent pas l'objet traité). Les moyens statistiques sont réapparus justement à cause des difficultés rencontrées par les moyens syntaxiques basés sur les grammaires formelles.

Cadres théoriques actuels des TAL

Ces cadres théoriques sont principalement syntaxiques.

cadres théoriques presque tous issus des travaux de Noam Chomsky

Rappelons les grandes lignes du programme chomskien:

- fonder une théorie de la compétence du locuteur natif, dans le cadre d'une épistémologie popperienne;
- les règles syntagmatiques de génération sont à la fois les hypothèses et l'outil de déduction dans la démarche hypothético-déductive: la génération par réécriture constitue le processus de déduction;
- les règles syntagmatiques modélisent des structures profondes et non des structures de surface;
- on reste dans le cadre de la phrase, le plus souvent simple (proposition principale, éventuellement une subordonnée), sans grand rapport (en complexité et longueur) avec les phrases réelles des textes: en fait des phrases artificielles, abstraites, expérimentales, produites par la génération;
- étant donné une grammaire générative, une phrase qui peut ou ne peut pas être générée, est dite grammaticale ou pas;
- recherche fondamentale pure sans préoccupation de TAL, ni TAL à but opératoire, ni TAL à but de confrontation entre concepts et objet réel.

légère influence de Lucien Tesnière

Les grammaires et les arbres de dépendance ont été importés dans les TAL aux débuts de la traduction automatique (cf. [Tesnière 59]), car la langue source était le russe, et Lucien Tesnière avait écrit des grammaires du russe qui furent utilisées au CETA (puis GETA) à Grenoble.

Inadéquation linguistique des cadres théoriques actuels des TAL

Sans remettre en question l'adéquation de ces cadres théoriques **linguistiques** aux objectifs de leur concepteur, nous allons tenter de montrer leur inadéquation aux TAL syntaxiques:

	cadres linguistiques	TAL syntaxiques
différence des objets:	compétence	langue concrète (≠ performance)
=> travail sur corpus:	aucun	indispensable
différence des objectifs:	modélisation théorique d'un objet statique	objectif opératoire = processus (dynamique)
opposition des démarches:	génération: phrase sortie	analyse: phrase en entrée

L'objet traité en TAL n'est pas la compétence ("la **connaissance** que le locuteur-auditeur a de sa langue", cf. [Chomsky 71], page 13), ni la performance ("l'**emploi** effectif d'une langue dans des situations concrètes", cf. [Chomsky 71], page 13), mais la part concrète, matérielle d'une langue: texte ou signal de parole en tant qu'**objet concret, matériel**, extérieur à l'humain.

Les grammaires formelles constituent un outil adapté à certaines fonctions: modélisation des structures profondes de la compétence du locuteur, modélisation de la syntaxe d'un langage de programmation, conception de compilateur, description d'un langage formel. Mais c'est un outil mal adapté à la description d'une langue en tant qu'objet concret, extérieur à l'humain, car une langue concrète a très peu de caractéristiques communes avec un langage formel (comme un langage de programmation).

Nous allons illustrer ce point central:

caractéristiques communes:

Ce sont tous deux des codes au sens large (des "languages"), des systèmes de signes conventionnels;
ils ont tous deux:

- un ensemble de formes possibles (sonores ou visuelles)
- une syntaxe = les règles d'organisation des formes
- une sémantique = les correspondances conventionnelles entre les formes et les sens.

quelques caractéristiques différentes:

critère	part concrète d'une langue	"langage" de programmation
exemple	français, chinois	ALGOL, Pascal, C, Lisp
origine	sociétés humaines, mères	quelques personnes
utilisation	communication entre humains	un humain commande le processeur
diachronie	existe: évolution	n'existe pas: figé
lexique	ouvert, évolutif	clos, figé
morphologie	flexion, dérivation, composition	formes invariables
syntaxe	peu connue, évolutive	totalement explicite, figée
redondance des formes	grande, peu connue	nulle ou artificielle, connue
segments récursifs	non	oui, sans limite
insertion des segments	insertion multiple limitée	insertion multiple non limitée
corresp. forme/sens	non biunivoque	biunivoque: 1 forme <-> 1 sens
conventionnalité	implicite	explicite
explicitation	incomplète	totale (ex.: anaphore inexistante)
métalangage	une langue	une langue aussi

Développons les aspects liés à la syntaxe:

La redondance des formes est une caractéristique des langues, comme de tout code utilisé par des êtres vivants (code génétique par exemple); elle permet une robustesse de la transmission et de la mémorisation des informations; comme un langage formel n'est pas redondant, une grammaire formelle n'est pas appropriée à tirer parti de cette redondance, qui constitue pourtant un des fondements des TAL.

La récursivité des segments (et donc des règles) est une hypothèse sur les structures profondes de la compétence du locuteur natif, mais elle n'est pas indispensable pour modéliser la syntaxe des langues concrètes, car il n'y a jamais une infinité de compléments, ni des insertions multiples illimitées, alors qu'elle est indispensable pour modéliser la syntaxe d'un langage de programmation, car il n'y a pas de limite a priori à l'enchâssement des instructions.

La polycatégorie (inexistante dans les langages formels), ou le fait qu'une même graphie recouvre plusieurs rôles syntaxiques et plusieurs sens est une conséquence de la correspondance forme-sens non biunivoque dans les langues; c'est un phénomène lexical hors-contexte, du point de vue du dictionnaire, mais il n'a plus d'existence en contexte.

Ce qu'on appelle d'habitude "ambiguïté" est en fait l'incapacité de choisir pour la machine qui analyse (en accord avec [Rady 83]), à cause du manque d'informations sur l'objet traité, du manque de contexte, mais *ce n'est pas une propriété intrinsèque de l'objet traité* (contrairement à [Rady 83]): cette "ambiguïté" est un artefact dû à l'absence de contexte, et cette incapacité de choisir entraîne une combinatoire artificielle; réservons plutôt à "ambiguïté" le sens d'une hésitation entre plusieurs interprétations pour l'humain qui écoute ou lit.

Quand on assigne à un analyseur l'objectif d'accepter ou de refuser une phrase pour diagnostiquer sa grammaticalité selon une grammaire formelle, on le place dans la position du locuteur natif qui juge de l'acceptabilité d'une phrase générée: ce n'est pas le rôle qu'on attend d'un analyseur.

Assignons plutôt à un analyseur l'objectif de produire une analyse pour toute phrase entrée.

On constate enfin l'absence de cadre théorique pour décrire et modéliser (et traiter) les formes des phrases de complexité syntaxique et de longueur telles qu'on les observe dans des textes attestés (avec des propositions subordonnées, antéposées ou incises, des verbes antéposés, des coordinations, des subordinations nombreuses et variées).

Inadéquation algorithmique des cadres théoriques actuels des TAL

On peut observer deux types de causes d'inadéquation algorithmique:
- les causes externes: l'inadéquation linguistique implique une inadéquation algorithmique: les algorithmes utilisent peu et mal les propriétés de l'objet réellement traité (propriétés des langues concrètes)
- les causes internes: les algorithmes fonctionnent par analogie avec la compilation, et sont fondés sur les propriétés (connues exhaustivement) des langages formels, alors qu'on traite des langues concrètes (propriétés connues partiellement)

Ceci entraîne que les algorithmes sont combinatoires: des *choix locaux* sont faits (presque) à l'aveuglette, car avec peu de critères de choix (ces critères ne peuvent être que d'origine linguistique); on a uniquement un critère de *diagnostic global* en fin d'analyse: cette chaîne est ou n'est pas une phrase, selon la grammaire formelle.

Voici les causes de la combinatoire des algorithmes, c'est-à-dire les caractéristiques des choix locaux:
- polycatégorie (inexistante dans les langages formels) => algorithme combinatoire, car on n'a pas de critère pour choisir la catégorie (propriété de langue inexistante dans un langage formel);
- syntagme récursif (non motivé dans les langues concrètes) => algorithme combinatoire, car, le syntagme incluant ses compléments, on segmente et on relie simultanément, sans critère pour borner le syntagme ni pour choisir les rattachements (relation = propriété de langue inexistante dans un langage formel)
- grammaire formelle => algorithme combinatoire, car plusieurs règles sont applicables simultanément, sans critère pour choisir quelle règle
- syntagme récursif dans une grammaire formelle => algorithme combinatoire, car on n'a pas de critère pour choisir combien de fois appliquer une règle récursive
- pas (ou peu) d'exploitation du contexte (règles hors contexte), ni de la redondance des formes.

Un algorithme combinatoire implique un processus arborescent: à chaque nœud, essai d'un choix-branche, échec éventuel, diagnostiqué à une feuille, => retour en arrière, pour essayer les autres branches de chaque nœud; l'algorithme est de complexité au mieux quadratique en temps.

Un algorithme combinatoire implique la sortie de zéro ou plusieurs analyses sans critère pour choisir.

De plus, l'unification, souvent utilisée, est aussi un processus combinatoire: l'ordinateur trouve des correspondances en essayant toutes les correspondances possibles, par manque de connaissances sur l'objet traité.

Notons qu'une grammaire formelle utilisée en analyse syntaxique automatique remplit simultanément deux fonctions: elle guide le processus d'analyse par l'application des règles (par réécriture en sens inverse de la génération), et elle code les structures des syntagmes (dans le membre droit des règles).

Enfin, on fait souvent l'hypothèse implicite mais erronée que tout l'objet analysé est connu (comme pour un langage formel): tous les mots, toutes leurs catégories possibles, toutes les structures: ces attendus sont irréalistes et imposent des rattrapages par des procédures ad hoc (qui exploitent enfin contexte et redondance).

Point de vue historique

Pourquoi les grammaires formelles ont-elles pris une telle importance en TAL?

Chomsky a eu une formation initiale en mathématique, puis il a importé en linguistique son bagage mathématique: les grammaires formelles sont nées en linguistique, sur un substrat mathématique.

ALGOL 60 est le premier langage de programmation décrit par une grammaire formelle.

Dans les années 60, Bernard Vauquois (qui faisait partie des créateurs d'ALGOL) fonde la traduction automatique *de langues* de seconde génération sur la compilation, traduction automatique *de langages formels* (voir [Boitet 92] page 50).

Du point de vue historique, le TAL reçoit donc les grammaires formelles de 3 origines différentes: la syntaxe chomskienne (linguistique), la compilation (informatique), et la traduction automatique (pratique fondatrice du TAL).

Dérive actuelle de la recherche vers l'activité de l'ingénieur

Le chemin de l'efficacité opératoire passe par la recherche fondamentale; or, depuis les débuts de la traduction automatique, il y a une tendance en TAL à vouloir faire l'économie de ce détour, ce qui explique en partie les difficultés actuelles.

Il s'agit plutôt de maintenir la spécificité et la complémentarité des 2 fonctions:

- recherche: produire de nouvelles **connaissances** sur l'objet étudié, puis les transmettre,
- activité de l'ingénieur: concevoir des produits (objets ou services) commercialisables.

L'ingénieur a besoin des résultats de la recherche fondamentale. Pour le chercheur, les nouvelles connaissances qu'il a produites sont validées du point de vue opératoire dans la conception d'un produit qui applique ces connaissances.

Une solution alternative *à l'inadéquation linguistique et algorithmique des cadres théoriques des TAL syntaxiques*

Remarques préliminaires

On présente dans cette seconde partie des principes linguistiques et algorithmiques permettant de construire des cadres théoriques appropriés aux TAL syntaxiques; on entend par "cadres théoriques appropriés aux TAL", le fait qu'ils permettent de concevoir des **algorithmes non combinatoires** et capables de traiter des textes attestés avec efficacité, robustesse et qualité; ces principes constituent une solution alternative, sans préjuger de l'existence d'autres solutions; cette "obligation de résultat" constitue un indice qu'un certain niveau théorique linguistique est atteint.

En préliminaire, on se base sur la constatation que la part concrète d'une langue (l'objet à traiter) n'est pas un langage formel (ce qui n'empêche pas de formaliser, représenter, coder les structures, les processus de traitement et de déduction), et on fonde la démarche sur l'étude concrète des corpus.

Principes de construction d'un cadre théorique linguistique

Comment fonder une syntaxe des langues concrètes?

Un premier point fondamental est de circonscrire correctement l'**objet**: objet réel, part concrète d'une langue, extérieure à l'observateur, ce qui implique l'étude des corpus, des textes et non pas des phrases isolées, des textes réels et non pas des phrases artificielles; l'objet est restreint aux formes, en excluant a priori les points de vue sémantique et cognitif (ce qui n'empêchera pas de rencontrer ces domaines à leurs frontières).

Quelle **méthode** utiliser?

On s'inspire des méthodes expérimentales utilisées en physique, en biologie; plus précisément, on utilise une méthode hypothético-déductive associée à l'induction: les hypothèses sont bâties par induction à partir de l'observation des corpus, sans a priori, mais par découverte des propriétés des corpus. Dans le cadre de cette méthode, l'ordinateur prend la place d'outil pour observer l'objet, expérimenter les hypothèses, confronter ces hypothèses à l'objet, ce qui implique l'analyse automatique, prenant en entrée l'objet à observer, à théoriser.

La syntaxe est conçue comme une science expérimentale; elle est définie comme science des formes des langues concrètes: quels sont les segments? quelles sont les relations entre segments? leurs propriétés, leur typologie, l'aspect topologique et géométrique: relation entre structure et ordre linéaire.

On expose maintenant une démarche de construction d'une syntaxe des langues concrètes en 3 étapes successives en distinguant conceptuellement et chronologiquement la définition des **segments** et la définition des **relations** entre segments:

1) définir des **segments** non récursifs et hiérarchisés:

. des segments *non récursifs* ne contiennent pas leurs compléments (segmenter sans relier);

. des segments non récursifs *hiérarchisés* ont une structure qui s'exprime en fonction des segments du niveau hiérarchique inférieur (de nature *différente*, sinon: même niveau => segment récursif);

. donc: dans des segments non récursifs hiérarchisés, un tout est de nature *différente* de ses parties.

Les niveaux de la hiérarchie des segments non récursifs sont: mot, séquence, bloc, phrase

les **séquences** sont constituées de mots autour d'un nom ou autour d'un verbe

. séquences: segments longs, mots centraux en liste ouverte (nom, adjectif, et verbe, adverbe), segments homogènes du point de vue de leur flexion (mêmes genre, nombre, personne, cas)

. clips (prépositions, conjonctions, pronoms relatifs, ponctuation): segments courts, mots très fréquents, en liste close (environ 30% des mots d'un texte)

. clips + séquences (nominales + verbales) constituent une double bipartition de la phrase

les **blocs** sont constitués de 0 ou 1 clip (`c`), suivi de 1 à 3 séquences (`N` = nominale, `V` = verbale)

. les structures de bloc sont en nombre restreint: `N NV NVN V VN cN cNV cNVN cV cVN`

. les blocs subordonnés ou coordonnés commencent par un clip

. insertion des blocs: un bloc inséré coupe un bloc entourant, entre 2 séquences: `[N [cN] VN]`
(définition topologique, non fondée sur une relation linguistique entre segments)

les **phrases** sont constituées de blocs contigus ou insérés

2) définir une **relation** linguistique entre segments:

la dépendance de détermination est définie comme la relation d'un segment en train d'être lu ou entendu, avec un segment déjà en mémoire (elle est distincte de la dépendance actancielle):

la phrase est vue comme entité à 2 faces: la face concrète dans l'ordre linéaire, la face abstraite dans l'arbre de dépendance (nœud = séquence); les 2 faces sont liées par des correspondances topologiques et géométriques: l'ordre linéaire code l'arbre de dépendance sous la forme du résultat d'une linéarisation optimisée sous des contraintes topologiques, géométriques et perceptuelles;

nœud à 1 dépendant => contiguïté entre régissant et dépendant (environ 70% des dépendances);

nœud à 2 dépendants => non-contiguïté entre régissant et un des 2 dépendants (le deuxième);

on peut faire l'inventaire exhaustif des nœuds à 2 dépendants ou plus: il y a 4 cas de dépendances entre séquences **non** contiguës:

1. bloc inséré dans un autre bloc
2. séquences coordonnées
3. blocs antéposés au bloc principal
4. structure actancielle de séquence verbale

3) définir un **segment** composé de **segments reliés contigus**:

la chaîne de séquences dépendantes contiguës constitue un groupe prosodique, car la coupure entre 2 chaînes, coupure entre 2 séquences contiguës **non** dépendantes est une coupure prosodique (pause et profil particulier de la fréquence fondamentale): *la structure prosodique de la phrase est une image de l'arbre de dépendance linéarisé*; on peut faire l'hypothèse que la structure prosodique permet à l'auditeur de reconstituer l'arbre de dépendance.

Principes d'algorithmique du TAL syntaxique

Ils sont fondés sur le cadre théorique linguistique.

On pourra éviter toute combinatoire au moyen de connaissances plus approfondies sur l'objet traité:

- donner une place centrale au concept linguistique de séquence (≡ catégorie générique)
- utiliser des segments non récursifs et hiérarchisés: séparer segmenter et relier, et traiter par couche en montant dans la hiérarchie et dans l'abstraction: mots puis séquences
- utiliser les indices contextuels locaux d'abord, les indices lexicaux ensuite (les dictionnaires deviennent secondaires)
- n'avoir aucun attendu global sur la structure d'une phrase

On cherche à caractériser le processus d'analyse par des déductions locales fondées sur des propriétés linguistiques *stables*, tout en ayant le moins possible d'attendu sur l'objet analysé.

Les algorithmes sont non combinatoires, donc de complexité linéaire en temps par rapport au nombre de mots de la phrase traitée, ce qui implique que le mot est traité en un temps constant.

Comme les algorithmes sont non combinatoires, il sort toujours une analyse et une seule.

Pour éviter la polycatégorie, on affecte à chaque mot **une seule** étiquette par défaut ("le" est déterminant par défaut); cette étiquette unique permet alors de faire des déductions sur les mots voisins ("dans" est suivi d'une séquence nominale), et elle peut être elle-même modifiée par déduction à partir des mots voisins ("le" est un pronom objet après un pronom sujet ou après la négation "ne").

Chaque déduction doit être faite à un niveau hiérarchique adéquat, au moment où on a les informations nécessaires: par exemple, on ne peut décider si "de" est préposition ou partitif au niveau des mots, mais au niveau des séquences, on peut savoir si la valence sujet ou objet d'un verbe est saturée ou pas; une valence non saturée peut alors le devenir par un sujet ou un objet précédé de "de" partitif.

Les algorithmes sont conçus dès le départ dans le cadre de connaissances incomplètes sur l'objet traité: lexique incomplet (mots grammaticaux et verbes suffisent) et structures incomplètes.

On utilise une propagation de règles déclaratives interprétées, écrites dans des formalismes conçus à partir des nécessités de l'analyse (et non a priori), qui peuvent être différents à différentes étapes.

Certains de ces principes sont déjà utilisés partiellement dans certains travaux récents (ceux de Jean-Pierre Chanod, ceux de Eva Ejerhed, et ceux de Atro Voutilainen par exemple).

Voici la démarche générale d'un algorithme fondé sur ces principes:

- sur la phrase représentée en mots: étiquetage des mots et **délimitation des séquences** par propagation de déductions locales sur quelques mots contigus (et non pas par reconnaissance de patrons); à cette étape, la difficulté est de choisir dans quelle chronologie on imbrique l'utilisation d'informations d'origine lexicale (mots grammaticaux, radicaux de verbes, règles sur les finales) et l'utilisation d'informations d'origine contextuelle (contexte des mots grammaticaux, contexte des formes verbales);

- sur la phrase représentée en séquences: **mise en relation des séquences** par un processus fondé sur des états de la mémoire: retenir sélectivement une séquence A (par exemple comme sujet en attente de verbe), s'en souvenir sélectivement en rencontrant une séquence B (par exemple une séquence verbale conjuguée), relier B à A, puis oublier sélectivement A (ce sujet n'attend plus de verbe), et oublier toutes les séquences situées entre A et B (propriété topologique de l'arbre linéarisé: oublier une branche terminée, car ces séquences n'attendent plus aucune relation).

Le segment pivot de la démarche est la **séquence**: d'abord délimitée, puis mise en relation avec d'autres séquences, comme nœud de l'arbre de dépendance, et unité de la métrique des dépendances.

État de la réalisation

Les grandes lignes du cadre théorique et des algorithmes sont décrites globalement dans [Vergne 94].

L'analyseur fonctionne sur le français (analyse de texte quelconque) et sur l'anglais ou l'espagnol (analyse d'un corpus): seules les données linguistiques changent, et sont choisies automatiquement d'après le texte entré (méthode de [Giguet 95]).

Les 30 phrases du test coordonné par Anne Abeillé dans T.A.Informations, volume 32, 1991, n°2 (pages 107 à 120), sont toutes analysées (toutes car l'algorithme est non combinatoire) et correctement étiquetées, alors que les 6 analyseurs testés avaient analysé de 14 à 20 phrases (Exploratexte de Machina Sapiens en avait analysé 27).

Un ouvrage en cours de rédaction contiendra la description détaillée du cadre théorique et des algorithmes.

Les règles de traitement de l'espagnol sont en cours de mise au point, et on constate que les règles de mise en relation des séquences sont pratiquement les mêmes dans les trois langues, ce qui est le signe qu'un certain niveau de généralité multilingue est atteint.

Bibliographie

Boitet Christian: *TA et TAO à Grenoble... 32 ans déjà*, t.a.l. (revue de l'ATALA), volume 33, numéro 1-2, Klincksieck (Paris), 1992

Chanod Jean-Pierre, **Tapanainen** Pasi : *Tagging French -- comparing a statistical and a constraint-based method*, EACL-95 (7th Conference of the European Chapter of the Association for Computational Linguistics), Dublin, mars 1995

Chomsky Noam: *Aspects de la théorie syntaxique*, Seuil, Paris, 1971, traduction en français par Jean-Claude Milner de: *Aspects of the Theory of Syntax*, MIT (Cambridge, USA), 1965

Ejerhed Eva: *Nouveaux courants en analyse syntaxique*, t.a.l. (revue de l'ATALA), volume 34, numéro 1, Klincksieck (Paris), 1993

Fuchs Catherine et **Victorri** Bernard: *Continuity in Linguistic Semantics*, Benjamins (Amsterdam), 1994

Giguet Emmanuel: *Multilingual Sentence Categorization According to Language*, communication à la conférence internationale "From texts to tags: issues in multilingual language analysis" (EACL SIGDAT workshop), Dublin, mars 1995

Gosselin Laurent: *Sémantique de la Temporalité: Un modèle calculatoire et cognitif*, habilitation à diriger des recherches, université de Caen, janvier 1995

Rady Mohamed: *L'ambiguïté du langage naturel est-elle la source du non-déterminisme des procédures de traitement?*, thèse de doctorat d'état, université de Paris 6, juin 1983

Tesnière Lucien: *Eléments de syntaxe structurale*, Klincksieck (Paris), 1982 (1ère édition: 1959)

Vergne Jacques: *A non-recursive sentence segmentation, applied to parsing of linear complexity in time*, communication et démonstration à la conférence internationale "New Methods in Language Processing 94" (NeMLaP 94), UMIST, Manchester, septembre 1994

Voutilainen Atro: *A syntax-based part-of-speech analyser*, EACL-95 (7th Conference of the European Chapter of the Association for Computational Linguistics), Dublin, mars 1995

•