Le développement de grammaires électroniques processus, modèles et outils

Mahmoud Fawzi MAMMERI
Université Blida 2

Résumé

Cet article décrit le processus de développement de grammaires électroniques et les outils qui s'y rapportent. Ce processus fait intervenir: la linguistique descriptive pour décrire et analyser un phénomène de langue, ensuite la linguistique formelle pour modéliser le phénomène déjà décrit et enfin la linguistique computationnelle pour implémenter, valider et évaluer les descriptions modélisées. Chacun de ces domaines proposent un ensemble d'outils. Le choix des outils adéquats dépend en premier lieu du cadre théorique adopté par le concepteur et qui peut être les grammaires de dépendance, les grammaires de constituants, etc. En optant pour les grammaires de constituants, par exemple, nous aurons à choisir entre l'hypothèse DP et l'hypothèse NP. L'article met l'accent sur certain outils actuellement proposés dans le cadre de l'analyse NP : les critères de Zwicky, les grammaires HPSG et le système LKB.

Mots clés:

linguistique computationnelle - grammaires électroniques - système LKB - HPSG.

الملخّص

يحـاول هـذا المقـال الكشـف عـن عمليـة تطويـر الأنحـاء الإلكترونيـة والأدوات المتعلقـة بهـا. تحتـاج هـذه العمليـة إلى ثلاثـة أصنـاف مـن الأدوات: أولا، أدوات لسـانية تسـمح بوصـف وتحليـل الظاهـرة (أو الظواهـر) اللغويـة المدروسـة، ثانيـا، أدوات صوريّـة مـن أجـل التعبـير عـن المعـارف اللسـانية المتحصـل عليهـا في إطـار نمـوذج يسـمح بالمعالجـة الآليـة، ثالثـا، أدوات حاسـوبية تسـتعمل توصيفـات المعـارف الصوريـة المتحصـل عليهـا في إطـار النمذجـة داخـل برامـج حاسـوبية حقيقيـة. ويتوقـف اختيـار الأدوات المناسـبة بالدرجـة الأولى عـلى الإطـار النظـري الـذي يعتمـد عليـه مطـورو النحـو الـذي يمكـن أن يكـون أنحـاء التبعيـة أو أنحـاء المكونـات ... كـما أنّ اختيـار أنحـاء المكونـات مثـلا يفـرض الفصـل في الاختيـار بـين مقاربتـي DP وNP في تحليـل المركـب الاسـمي. في هـذا المقـال ركزنـا عـلى بعـض الأدوات المقترحـة حاليـا في إطـار افـتراض NP وهـي مقاييـس Zwicky ونحـو HPSG ونظـام LKB.

الكلمات المفاتيح:

لسانيات حاسوبية - نحو حاسوبي - نظام BKL - GSPH.

Abstract

This paper describes the computational grammar development process and related tools. Tree sub domains of linguistics are involved: descriptive linguistics which is necessary for analysis and description, formal linguistics for modeling, and computational linguistics for implementation, validation and evaluation. Each of these domains proposes a number of tools. Selection of suitable tools depends on the theoretical framework adopted by the grammar designer: dependency grammars, phrase structure grammars etc. The choice of phrase structure grammars, for example, signifies that we have to choose between using DP hypothesis and NP hypothesis. The article emphasizes on number of tools currently used in NP analysis: Zwicky criteria, HPSG grammar and the LKB system.

Keywords:

Computational linguistics - computational grammars - LKB system - HPSG

1. Introduction

Le développement de grammaires électroniques se fait en trois étapes distinctes, chacune faisant appel à un domaine de la linguistique. La première étape recours à la linguistique descriptive pour décrire et analyser les phénomènes de langue à intégrer dans la grammaire. Dans la deuxième étape, le développement de grammaires électroniques a besoin de la linguistique formelle pour modéliser et représenter les phénomènes déjà décrit. Enfin, il s'agit dans une troisième étape d'implanter et de valider les descriptions (ou grammaires) adoptées dans les deux premières étapes dans un système de performance[1], ce qui nécessite le recours à la linguistique computationnelle. L'extrant d'une étape est l'intrant de la suivante. L'extrant de la dernière va permettre une rétroaction (feedback) : le résultat de l'implantation sert à revenir aux deux premières étapes pour confirmer ou infirmer les hypothèses ayant servi pour l'analyse et la modélisation et, si nécessaire, apporter les modifications qui s'imposent. Le fil conducteur entre ces trois étapes — et ces trois domaines de la linguistique — est le phénomène linguistique à traiter. Toutes ces étapes — qui constituent le processus de développement de grammaires électroniques — nécessitent des moyens de bord : des modèles et des outils. Dans cet article, il sera donc question, d'une part, de lever le voile sur les modèles et/ou outils existants et, d'autre part, de se demander par quels moyens traiter au mieux les phénomènes rencontrés dans les langues naturelles. La linguistique étant un domaine assez large, les moyens qu'elle met à la disposition du chercheur sont très variés et dépendent généralement du cadre théorique adopté. Cela est tout aussi vrai pour la linguistique computationnelle. Pour s'en convaincre, il suffit de faire une revue sur des incontournables du domaine tels que Chomsky (1981), Kaplan et Bresnan (1982), Pollard et Sag (1994) et Joshi (1987). Ainsi, faudrait-il dès le départ dessiner le contour de cette question. Nous allons nous restreindre tout d'abord à la syntaxe et au lexique[2]. Ensuite, nous allons nous placer dans un cadre lexicaliste et endocentrique. Le lexicalisme veut dire, grosso modo, que « la syntaxe n'a pas accès à la construction interne du mot, en d'autres termes, elle n'a aucun droit de regard sur le lexique». (Anderson 1988:165) L'endocentrisme veut dire que « dans tout syntagme, il doit y exister une tête, i.e. une branche sœur qui domine toutes les autres». (Bloomfield, 1933:183)

Dans la section subséquente, nous donnons un survol sur le développement de grammaires électroniques. Nous passons en revue le domaine de prédilection de ces grammaires, le besoin qui est émit à leur égard, leur complexité, le processus suivi par le concepteur dans la démarche de construction de grammaires

ainsi que les types d'outils utilisés dans une telle démarche.

2. Développement de grammaires électroniques: motivations et processus

2.1 Place de la syntaxe et du lexique

Le développement de grammaires électroniques repose sur un certain nombre de phénomènes issus de la langue objet de l'étude. Ces phénomènes peuvent appartenir à différents niveaux de l'analyse linguistique (morphologique, syntaxique, sémantique, pragmatique, etc.) et même à plusieurs à la fois (morpho-syntaxe, syntaxe-sémantique, etc.). L'intérêt que nous portons au lexique et à la syntaxe a sa justification. Pour le lexique, il est claire qu'une grammaire, quelle qu'elle soit, doit pouvoir opérer sur un lexique. Quant à la syntaxe, elle devrait être un composant déterminant de toute application en TALN. Ainsi, la syntaxe formelle est devenue depuis plusieurs décennies un objet d'étude autonome distincte à la fois de l'étude du lexique et de celle du sens. Ceci se justifie par le fait que nous sommes passés de systèmes utilisant des informations syntaxiques très grossières à des approches permettant de décrire d'une manière très fine des phénomènes syntaxiques non triviaux (e.g. extractions, dislocations, etc.).

2.2 Ingénierie de grammaires: besoins et complexité

Exprimer les grammaires des langues naturelles de manière formelle est important pour au moins deux objectifs : l'un théorique, l'autre pratique. D'un point de vue théorique, les modélisations formelles sont devenues à l'heure actuelle un outil scientifique incontournable, et aucune science ne peut nier leur utilisation. En linguistique, en particulier, le recours aux grammaires formelles facilite la progression scientifique : il est plus facile de tester (i.e. confirmer ou infirmer) et comparer des hypothèses qui sont formulées d'une manière précise que de le faire avec des hypothèses formulées de manière élémentaire et vague. Ainsi, le développement même d'une petite grammaire avec l'aide d'un environnement de développement de grammaires peut assister le linguiste dans le test d'hypothèses et pour développer des analyses plus satisfaisantes. Ce type de rédaction de grammaires peut aussi aider à déterminer si des analyses de phénomènes variés sont compatibles, en d'autres termes, s'il est en réalité possible de les implémenter au sein d'une même grammaire. D'autre part, le développement de grammaire d'un point de vue applicatif est devenu de plus en plus important. Une grammaire formelle peut servir comme base pour des systèmes automatiques de traitement de la langue qui auront un impact considérable sur nos vies quotidiennes. La demande pour des grammaires avec une couverture raisonnable

pour des textes authentiques est devenue palpable dans plusieurs domaines différents, parmi lesquels la traduction automatique, l'extraction d'information et la création multilingue. Cependant, toute grammaire de ce genre couvrant plus d'un fragment trivial de la langue est énormément complexe et exige un travail coopératif de plusieurs spécialistes. Par conséquent, il y a un besoin pour une ingénierie et pour une réflexion sur les caractéristiques d'une bonne ingénierie.

2.3 Développement de grammaires électroniques: processus et outils

Le développement de grammaires électroniques nécessite trois catégories d'outils : des outils linguistiques, des outils formels et des outils computationnels. Les outils linguistiques, issus de la linguistique descriptive, servent à la description des diverses connaissances relatives au(x) phénomène(s) de langue étudié(s). En syntaxe, par exemple, il peut s'agir de théories, tests, critères, gloses, terminologies, …, qui permettent de développer une analyse du (des) phénomène(s) à traiter. Les connaissances linguistiques brutes résultant d'une analyse linguistique ne sont pas directement implantables dans un système informatique. Des outils formels, issus de la linguistique formelle, viennent alors pour répondre à un besoin de modélisation rigoureuse. Ces connaissances sont tout d'abord formalisées dans un modèle. Les raisons en sont simples : d'une part, les connaissances linguistiques ne sont généralement pas sous une forme directement exploitable par un ordinateur, et d'autre part, il y a un souci d'indépendance entre analyse linguistique — données et règles — et implantation informatique pour garantir une certaine liberté dans l'analyse. Ces modèles (ou formalismes) jouent ainsi en quelque sorte le rôle d'une interface entre une théorie et son implantation. Ces formalismes sont l'expression — en termes de représentation — de la théorie et en principe chaque théorie se dote d'un formalisme. Il s'agit donc d'outils formels nécessaires à l'expression des connaissances linguistiques dans un formalisme qui convient au traitement automatique. Enfin, et afin d'utiliser la description formelle des connaissances linguistiques dans un cadre informatique concret et l'appuyer par des grammaires computationnelles fonctionnelles, le concepteur de grammaire fait appel à des outils informatiques, issus de la linguistique computationnelle.

3. Modèles linguistiques modernes

Avant d'exposer les outils utilisés dans le processus de développement de grammaires électroniques, nous allons parler des modèles de la linguistique moderne qui sont à l'origine de ces outils. Faut-il le rappeler, il ne s'agit pas dans ce texte d'une présentation exhaustive des modèles et des outils disponibles dans la

littérature linguistique, mais plutôt d'une revue d'un sous ensemble de ces modèles et de ces outils qui sont compatibles les uns avec les autres et qui peuvent ainsi faire partie d'un même processus de développement de grammaires. Dans cette perspective, nous avons choisit un exemple d'un processus type qui est essentiellement basé sur le cadre théorique des grammaires dites d'unification puisque ces grammaires ont donné naissance à des implantations informatiques réelles.

3.1 Formalisation des représentations linguistiques

La volonté d'effectuer des traitements informatiques de productions langagières est à l'origine d'une expression mathématique des connaissances sur les langues. Dès le début du Traitement Automatique des Langues Naturelles (TALN), et pour résoudre la problématique de la traduction automatique, s'est posée la question de trouver des représentations sous format numérique pour des données linguistiques. C'est ce qui a amené, depuis 'Structures syntaxiques' de Chomsky, des chercheurs en linguistique à se doter d'une démarche scientifique comparable à celle suivie dans le modèle des sciences de la nature, ce qui a donné naissance à une multitude de formalismes syntaxiques. Ces formalismes ont ouvert une nouvelle ère pour la linguistique, et ont donné naissance à des applications informatiques très intéressantes. En parallèle, ont été développés des outils fondés sur les statistiques et les probabilités et sont utilisés dans des applications spécifiques de traitement automatique. Ces outils, même s'ils font appel à des concepts mathématiques, sont très éloignés d'une mathématisation des connaissances linguistiques:

«Prenons les modèles probabilistes markoviens : ils peuvent servir aussi bien à prédire le prochain mot que va prononcer un locuteur que le temps qu'il fera demain. Quant aux automates finis, très utilisés par les systèmes d'analyse robuste, leur inadéquation en tant que modèle de la syntaxe a été démontrée par Chomsky (1957). Quoi qu'il en soit, l'objet des travaux de TAL robuste n'est pas de construire des généralisations sur les langues». (Cori 2003: 17)

Dans ce qui suit, nous nous intéressons uniquement aux formalismes syntaxiques qui sont mathématiquement rigoureux.

3.2 Grammaires formelles et langages naturels

Les grammaires formelles sont des descriptions finies de langages particuliers. Leur grand intérêt est qu'elles permettent d'envisager le problème de la reconnaissance — i.e. décider si une séquence donnée du langage est bien formée — sans le recours à un examen exhaustif des chaînes du langage. Une

grammaire ne fournit pas uniquement une réponse binaire — bien formée vs. mal formée — au problème de la reconnaissance, mais elle assigne également des structures aux chaînes du langage — le plus souvent sous la forme d'un arbre syntaxique. Grâce à la récursivité, une grammaire permet d'engendrer, à l'aide d'un nombre fini de règles (de réécriture), la description structurale d'un nombre infini d'énoncés.

Chomsky identifie une hiérarchie de familles de langages de complexités croissantes, chaque famille correspondant à une contrainte sur la forme des règles de réécriture utilisées dans la grammaire. Sa grammaire la plus célèbre est la CFG (Context Free Grammar, en français Grammaires à Contexte Libre). C'est cette grammaire qui a suscitée beaucoup de travaux et critiques, et c'est elle qui a été la première à être utilisée pour modéliser et traiter les phénomènes linguistiques. Les CFG-s ont connu par la suite une large exploitation par la communauté du Traitement Automatique des Langues. Exploitées comme telles, au départ, pour développer les algorithmes d'analyse syntaxique des années 1960, très utilisés dans les langages formels, elles ont été adaptées et reprises par la majorité des formalismes développés par la suite.

3.3 Les courants non chomskyens

Des courants autres que celui développé par Chomsky — i.e. se basant sur les CFG-s et la représentation syntaxique d'une phrase à l'aide d'un arbre syntaxique — ont de leur côté participé au développement de modèles mathématiques du langage, modèles qui ont évolué jusqu'à aujourd'hui. Il s'agit des grammaires de dépendance de Hays[3] (1960, 1964) et les grammaires catégorielles de Bar-Hillel. Cependant, l'engouement extraordinaire pour les grammaires génératives-transformationnelles de Chomsky dans les années 60–70 a retardé l'essor de ces deux types de grammaires qui prouvent actuellement leur bonne adéquation au traitement automatique des langues. La grammaire de dépendance est qualifiée de plus apte au traitement des langues dans lesquelles l'ordre des mots est plus ou moins libre. Ce formalisme permet aussi de représenter la sémantique d'une phrase. Malgré que les grammaires de dépendance soient qualifiées comme diamétralement opposées aux grammaires de constituants, elles sont à placer, en termes de puissance d'expression, au même rang que les CFG. Gaifman (1965) démontre leur équivalence faible : « tout ce qui peut être décrit par une grammaire hors contexte peut être décrit par une grammaire de dépendance et vice-versa, mais les structures construites ne sont pas identiques. » Dans la suite du document, toutes les grammaires que nous allons rencontrer prennent

leurs sources dans la grammaire générative de Chomsky.

3.4 Lexicalisation et modularisation

Deux caractéristiques importantes ont principalement caractérisés les théories linguistiques modernes: lexicalisation et modularisation. Elles ont permis une plus grande précision dans la représentation et dans le contrôle de l'information linguistique. La première caractéristique est au premier plan grâce à un lexique contenant de plus en plus d'informations variées (morphologiques, syntaxiques, sémantiques, …). Ce courant de lexicalisation a pour contrepartie une meilleure généralisation des principes d'analyse décrits : si les informations locales ou spécifiques sont contenues par le lexique, il devient possible de représenter à un niveau très général les informations pouvant porter sur des langues ou des ensembles de langues. C'est ce que se proposent de faire les théories lorsqu'elles utilisent la notion de principes universels. Dans cette perspective, nous pouvons ainsi formuler des principes très généraux qui s'appliquent aux objets linguistiques manipulés tandis que les propriétés spécifiques sont quant à elles contenues dans le lexique (Blache, 1999: 21). La deuxième caractéristique, la modularisation, découle de la première, puisque la lexicalisation permet une meilleure structuration de l'information et représente de façon autonome des informations qui, dans les approches classiques, peuvent être soit implicites, soit représentées conjointement à d'autres propriétés. Il s'agit donc, plutôt, de dégager des principes universels qu'à distinguer les particularités de chaque langue, et cela ouvre la voie, d'un point de vue applicatif, au développement de grammaires électroniques multilingues.

4. Outils pour le développement de grammaires électroniques

4.1 Outils de la linguistique descriptive

Les outils de la linguistique descriptive sont nombreux et divers et nous ne pouvons les discuter dans ce texte. Pour s'en convaincre, il suffit d'essayer de les énumérer. Ces outils sont utilisés pour décrire, analyser, comparer, organiser, exprimer, …, les phénomènes de langues choisis. Par exemple, pour une analyse du syntagme nominal, on pourrait se placer dans l'un ou l'autre des deux cadres théoriques les plus populaires: les grammaires de constituants ou les grammaires de dépendance. Le fait d'opter pour les grammaires de constituants, par exemple, offre la possibilité de choisir entre trois types d'analyse: l'analyse DP (e.g. Abney, 1987), l'analyse NP (e.g. Pollard & Sag, 1994) et les analyses dites mixtes (e.g. van Eynde, 2006)[4]. Là encore, le choix de l'approche de l'analyse NP nous met en face de deux types d'outils, chacun d'eux est lié à

une hypothèse fondamentale. Le premier type d'outils est connu en littérature sous le nom de critères de Zwicky pour la capitalité (e.g. Zwicky, 1985c). Ces outils viennent en réponse à un besoin exprimé par l'hypothèse endocentrique (cf. supra) et qui consiste à élire une tête pour un syntagme donné. Le second type d'outils sont les critères de Zwicky pour l'affixalité (e.g. Zwicky, 1977). Ces critères répondent par ailleurs à une question fondamentale que nous pouvons paraphraser en «certains éléments de la chaine linéaire d'une phrase qui sont à la frontière d'autres éléments adjacents ont un statut qui n'est pas claire entre celui du mot, du clitique ou de l'affixe, quel serait alors leur statut réel ?» Ces critères sont invoqués lorsqu'il s'agit d'opter pour l'hypothèse lexicaliste (Chomsky, 1970) qui stipule grosso modo que «le lexique est imperméable aux opérations syntaxiques».

4.2 Outils de la linguistique formelle

Les outils de la linguistique formelle auxquels nous nous intéressons ont ceci de commun : ils appartiennent tous à une communauté de grammaires connues sous l'appellation «grammaires d'unification». Nous présentons tout d'abord ce type de grammaires dans la section subséquente. Ce sera aussi l'occasion de présenter le formalisme des structures de traits qui est à la base des descriptions issues de ces grammaires. Les grammaires d'unification qui ont suscitées le plus grand nombre de travaux en TALN sont la grammaire syntagmatique endocentrique ou HPSG (Pollard & Sag, 1994), la grammaire lexicale fonctionnelle ou LFG (Kaplan & Bresnan, 1982) et les grammaires d'arbre adjoint ou TAG (Joshi, Levy & Takahashi, 1975). Nous présentons dans la section 4.2.3 la grammaire HPSG. Certes, HPSG bénéficie d'un grand intérêt, visible à travers une communauté assez large par rapport aux autres modèles, mais sa présentation dans ce texte n'a qu'un intérêt purement didactique: l'outil computationnel que nous présentons dans la section 4.3.1, qui n'est autre qu'un système de performance, est une implantation de la grammaire HPSG. À la fin de ces différentes présentations, le lecteur aura une idée claire, mais surtout complète, sur tout le processus de développement de grammaires électroniques et pourra par la suite opter pour l'un ou l'autre des modèles de grammaire existants. Dans cet ordre d'idée, nous donnons dans la section 4.2.2 un certain nombre de critères communément utilisés pour choisir un modèle (outil) parmi plusieurs modèles (outils) concurrents.

4.2.1 Grammaires d'unification

Dès la fin des années 1970, de nouveaux modèles et théories linguistiques,

connus sous le nom de formalismes des grammaires basées sur l'unification, ont vu le jour. Il a été proposé de conserver les grammaires syntagmatiques tout en accroissant leur capacité descriptive. Il fallait enrichir ces grammaires de telle sorte qu'elles puissent permettre l'expression claire et distincte des principes organisateurs des langues. Ces formalismes grammaticaux fournissent ainsi des outils précis de description des langues naturelles pour décrire (i) l'ensemble des phrases possibles d'une langue (ou grammaticalité), (ii) les propriétés structurales de ces phrases (ou syntaxe) ainsi que (iii) leur signification (ou sémantique). (Shieber, 1990: 29) Ces formalismes ont atteint un niveau de maturité assez avancé, qui leur a procuré une place particulière dans le domaine de la linguistique d'une manière générale mais surtout dans trois de ses sous domaines: les linguistiques descriptive, formelle et computationnelle. Ces modèles recherchent une articulation explicite entre le lexique, la syntaxe et la sémantique: aux différents objets linguistiques (morphèmes, syntagmes, constructions) sont associées des informations (propriétés linguistiques) de différentes natures (lexicales, syntaxiques, sémantiques, …) qui seront combinées par des opérations variées, dont l'unification qui occupe une place centrale. Cette conception intégratrice est un atout majeur pour le traitement automatique des langues. De plus, les modèles sur lesquels se basent ces grammaires sont des modèles logiques et/ou mathématiques (grammaires de constituants, structures de traits) solides et pour lesquels ont été définies des méthodes de programmation. (Abeillé, 2007: 25–6)

4.2.1.1. Intérêts des grammaires d'unification pour le traitement automatique des langues

L'intérêt des grammaires d'unification pour le TALN est triple: d'une part, en tant que systèmes formels, (i) elles facilitent la définition d'algorithmes d'analyse, d'autre part, (ii) elles permettent l'intégration de ressources hétérogènes, à la fois parce qu'elles utilisent le même type de représentation (les structures de traits) et parce qu'elles permettent l'accumulation harmonieuse de contraintes sans leur imposer d'ordre d'application, et enfin, par la séparation stricte qu'elles établissent entre données linguistiques et programmes de traitement, (iii) elles sont neutres quant à leur application (elles peuvent être utilisées en analyse comme en génération ou en désambiguïsation). (Abeillé & Blache, 2000: 58)

4.2.1.2. Caractéristiques communes aux grammaires d'unification

Les grammaires basées sur l'unification se rejoignent dans plusieurs points

essentiels, en particuliers: (i) une formalisation à base de structures de traits, (ii) le souci d'une articulation plus explicite du lexique, de la syntaxe et de la sémantique en adoptant le même type de représentation (structures de traits) et en manipulant des descriptions partielles spécifiées au fur et à mesure des combinaisons et selon les besoins, (iii) ces grammaires reposent sur des formalismes déclaratifs, donc aucun ordre dans l'application des règles, et monotones, dans le sens où chaque règle ne peut qu'ajouter de l'information, sans modification destructrice, et enfin, (iv) elles sont allées plus loin dans l'abandon de la notion de règle, les règles dynamiques de production sont en effet remplacées par des conditions statiques de bonne formation, d'où leur appellation alternative «grammaires basées sur les contraintes»[5] (Abeillé & Blache, 2000: 57–8).

4.2.1.3. Un nouveau mode de représentation: les structures de traits

La structure de traits (ou structure attribut-valeur) est un mode de représentation des objets linguistiques. Plusieurs approches différentes de la grammaire des langues naturelles ont élaboré cette notion comme outil de description et les grammaires d'unification en sont un exemple. Les structures de traits, bien qu'elles soient différentes dans le détail d'une théorie à une autre, ont toutes en commun de contenir des ensembles d'attributs, chaque attribut ayant la forme d'une paire <nom, valeur>. La valeur d'un attribut peut être un symbole atomique ou à nouveau une autre structure de traits enchâssée. Il existe plusieurs manières de représenter la structure de traits. La plus populaire — utilisée dans les grammaires HPSG et LFG — est celle utilisant des matrices attribut-valeur. Entre différentes structures attribut-valeur, ont été définies plusieurs relations dont l'extension et la subsomption et plusieurs opérations dont la négation, la disjonction, la généralisation et plus particulièrement l'unification. Cette dernière, qui a joué un rôle majeur dans les langages de programmation tels que Prolog, est également au centre des formalismes syntaxiques modernes. De façon informelle, l'unification de deux structures de traits correspond à leur fusion dans une seule structure éventuellement plus complète, puisqu'en plus des attributs qui sont spécifiés dans les deux structures de départ, et qui doivent être compatibles, la structure résultante contient tous les attributs qui apparaissent dans l'une ou dans l'autre des deux structures de départ. Et c'est justement cette opération d'unification qui permet, si elle réussie, l'accumulation d'information.

4.2.1.4 Orientation lexicaliste des grammaires d'unification

La plupart des formalismes basés sur l'unification (HPSG, LFG et les TAG en particulier) ont tendance à utiliser un style d'analyse lexicaliste. La puis-

sance de ces grammaires vient du fait que les structures lexicales manipulées sont porteuses de plus d'informations et sont donc plus complexes. Et c'est justement l'existence simultanée de différents types d'informations, notamment syntaxiques et sémantiques, au niveau des structures lexicales qui permet au lexique de jouer un rôle crucial dans l'analyse linguistique, en particulier dans l'interfaçage (e.g. syntaxe-sémantique). Dans certains modèles de grammaires, tel que les TAG, l'intégration du lexique et de la syntaxe peut aller jusqu'à la lexicalisation intégrale de la grammaire.

L'enrichissement du composant lexical par des informations syntaxiques notamment, codées dans les entrées lexicales sous forme de traits, s'est accompagné d'une réflexion sur l'organisation lexicale. La notion de règle lexicale, conçu sur le modèle des règles de flexion ou de dérivation définies en morphologie, a été étendue à la représentation de divers phénomènes syntaxiques (e.g. passif, constructions impersonnelles), qui sont analysés comme des relations entre des items lexicaux plutôt que comme des relations entre des phrases.

4.2.2. Quelques indices pour le choix d'un modèle

Le premier choix à effectuer dans la construction d'une grammaire est celui du formalisme grammatical employé pour la décrire. Le choix d'une théorie linguistique (par opposition à un formalisme ad hoc) permet de bénéficier des acquis des recherches linguistiques dans le traitement des problèmes syntaxiques variés, et d'en attendre des solutions générales et cohérentes. La déclarativité du formalisme est fondamentale si l'on veut développer une grammaire de large couverture. Un contre-exemple notoire est donné par les réseaux de transitions augmentés (ATN) dont les conditions et actions, par essence procédurales, deviennent rapidement inextricables. D'autre part, le choix d'un modèle est lié à la question de l'interface syntaxe-sémantique : la représentation syntaxique construite doit permettre un passage aisé à une représentation sémantique. Enfin, le langage naturel est ambigu, et cette ambiguïté peut appartenir à des niveaux différents de la représentation de la phrase. Le fait d'intégrer ces niveaux dans un même système — la structure de traits en est un exemple — permet d'éviter une explosion combinatoire.

4.2.3. Les grammaires HPSG: un exemple de choix de modèle

La grammaire syntagmatique endocentrique (aussi, grammaire syntagmatique guidée par les têtes) est arrivée à un stade relativement stable dans son évolution théorique: Pollard et Sag (1987), Pollard et Sag (1994), Sag et Wasow (1999) et Sag et al. (2003). Elle a connu au fil des années une évolution rapide

vers une plus grande précision de la représentation et du contrôle de l'information linguistique. C'est une théorie linguistique qui se propose de fournir un cadre de modélisation de principes grammaticaux universels. HPSG construit en même temps différents niveaux de représentation pour un énoncé dans un seul système de description qui est la structure de traits. Ce qui permet une articulation souple entre le lexique, la syntaxe et la sémantique. De plus, et dans un but de généralité et d'extensibilité, les connaissances linguistiques représentées sont décrites de façon déclarative et sont séparées des programmes les appliquant. HPSG a bénéficié des travaux de plusieurs cadres théoriques qui l'ont précédé dont particulièrement la grammaire LFG et les grammaires catégorielles (CG). HPSG s'est aussi inspirée, d'un point de vue plus formel, de travaux en logique et en informatique sur le typage et l'héritage. Toutes ces caractéristiques plaident pour le choix de la grammaire HPSG comme modèle de description.

4.3. Outils de la linguistique computationnelle

Dans la réalisation d'application en traitement automatique des langues et plus particulièrement dans le cadre du développement de grammaires, on peut recourir à l'utilisation d'un langage de programmation, ce qui constitue un environnement de travail de plus bas niveau. Dans ce cas, certains langages sont plus appropriés que d'autres pour une tâche donnée. Dans le cas de l'analyse syntaxique, il peut s'agir de Lisp ou Prolog par exemple. On peut aussi utiliser une plate-forme de développement spécialisée, intégrant des facilités et masquant à l'utilisateur des aspects spécifiques aux langages de programmation.

4.3.1. La plateforme LKB: un système de performance pour les grammaires HPSG

Différentes implantations ont été proposées pour le modèle HPSG. La plus utilisée est le système LKB (Linguistic Knowledge Building) (Copestake, 2002). LKB est un environnement de développement de grammaire et de lexique qui a été conçu pour les formalismes à base de contraintes. Il est spécialement conçu pour l'utilisation des structures de traits typées. Par conséquent, il est destiné pour l'implémentation de toute grammaire à base de structures de traits. Il est utilisé dans les recherches en traitement automatique des langues qui impliquent les formalismes linguistiques basés sur l'unification en analyse comme en génération. En gros, c'est un environnement de développement spécialisé de très haut niveau qui peut être utilisé pour le développement de différentes grammaires de différentes tailles (Copestake & Flickinger, 2000: 1). Le système LKB possède une combinaison de caractéristiques qui le distingue. Les plus importantes sont:

- Distribution libre. Il s'agit d'un environnement de développement de grammaires distribué comme partie des outils LinGO disponible en open source (http://lingo.stanford.edu).
- Richesse didactique. Il offre au développeur des grammaires à large couverture qui supportent interprétation sémantique et génération. Ainsi, plutôt que de concevoir la totalité d'une grammaire à partir de zéro, il serait plus intéressant de choisir de réutiliser une grammaire déjà existante et de l'adapter à ses besoins.
- Disponibilité en multiplateforme. Il est implémenté en Common Lisp, distribué, non seulement, comme source, mais également comme application autonome qui peut être exécutée sous Linux, Solaris et Windows et fonctionne aussi sous Macintosh Common Lisp (sous licence).
- Outils d'aide au développement. Il s'agit d'un atelier qui comprend (i) un analyseur, (ii) un générateur, (iii) des outils variés pour la manipulation des représentations sémantiques, (iv) un ensemble riche d'outils graphiques pour l'analyse et le débogage de grammaires, (v) une importante documentation en ligne et supporte (vi) une large gamme de hiérarchies d'héritage.
- Caractère multi-formalismes. Il s'agit en fait d'un cadre de développement de grammaires indépendant de tout formalisme. Des grammaires de toutes les tailles ont été écrites en utilisant le système LKB, et pour différentes langues, principalement avec les cadres linguistiques des grammaires catégorielles et les grammaires HPSG.

4.3.2. Autres outils complémentaires

En plus des outils relatés dans la section précédente, nous allons parler dans ce qui suit de trois autres outils très importants dans le développement de grammaires computationnelles. Ces outils sont associés aux systèmes de performance et plus particulièrement le système LKB. Ce sont les phrases de test[6] (ou suites de test, en anglais 'test suites'), la sémantique à récursion minimale (Minimal Recursion Semantics, ou MRS) et la Matrix. Ces outils ne sont pas des systèmes : (i) la MRS est un cadre (ou modèle) pour la sémantique, (ii) la Matrix est une grammaire (ou plutôt un noyau de grammaire) et (iii) les phrases-tests sont des données, sur un ou plusieurs phénomènes de langue, spécialement construites pour tester une grammaire. Pour situer ces outils par rapport au système de performance LKB, nous dirons que (i) les phrases-test sont prises en charge par LKB pour valider des grammaires, (ii) la MRS fait partie intégrante de LKB et sert à doter une grammaire d'un module sémantique et (iii) la Matrix est un

produit — une grammaire implantée — du système LKB dans lequel elle peut être paramétrée et réutilisée pour donner lieu à un nouveau produit — donc, une nouvelle grammaire — selon les besoins du développeur de grammaires.

4.3.2.1. Outils d'évaluation: compétence et performance

Les grammaires implantées peuvent être évaluées en conséquence selon deux dimensions : compétence et performance. La compétence d'une grammaire se rapporte à sa couverture et sa précision. En d'autres termes, sa capacité de rendre compte de toutes et rien d'autres que les phrases supposées être grammaticales. La performance se rapporte quand à elle aux ressources qui sont utilisées durant le traitement. Ces ressources sont essentiellement le temps processeur et l'espace mémoire alloué aux structures de données réclamées pour l'analyse.

L'évaluation des applications en TALN nécessite une méthodologie et des outils. Les jeux de phrases de test et les corpus de textes représentent les deux outils les plus utilisés pour ce type d'application. Elles peuvent être considérées comme complémentaires: l'intérêt des corpus est que leurs données sont «naturelles» tandis que l'avantage des jeux de phrases de test réside dans leurs caractères systématique et exhaustif. Cependant, il existe une autre propriété importante qui plaide en faveur de l'utilisation des phrases de test: celles-ci incluent aussi des données agrammaticales — absentes des corpus ! — qui s'avèrent être indispensables pour l'évaluation diagnostique (e.g. en termes de couverture, de surgénération ou d'efficacité). Le système LKB fournit une manière pour définir un jeu de phrases de test qui peut être utilisé comme test de performance (benchmarking facility). Une analyse en lot (batch parse) donne alors pour chaque phrase contenue dans le jeu de phrases de test le nombre d'analyses possibles ainsi que les tranches d'arbre passives (passive edges). En termes de performance, LKB donne le total CPU de toutes les phrases analysées.[7] Des outils plus sophistiqués pour l'évaluation de la compétence et de la performance des grammaires sont disponibles dans LKB à travers le package [incr tsdb()] (Oepen, 2001).

4.3.2.2. La sémantique MRS

La MRS (Minimal Recursion Semantics) (Copestake et al., 2005) est un cadre pour la sémantique computationnelle. C'est le formalisme standard utilisé à grande échelle dans les grammaires HPSG. MRS n'est pas, en soi, une théorie sémantique à part entière. Il s'agit d'un langage de description pour les formules de la logique du premier ordre (First Order Logic). Le système LKB contient un module pour l'analyse des représentations MRS. Ce module est indépendant du

reste du LKB et fournit des outils pour la manipulation des structures MRS dans des représentations en structures de traits (Copestake & Flickinger, 2000)

4.3.2.3. La LinGO Grammar Matrix

4.3.2.3.1. Une nouvelle approche pour le développement de grammaires

La production à grande échelle de grammaires basées sur les contraintes et d'environnements d'analyse adéquat est un processus qui fait l'objet d'un effort intense et à forte consommation de temps et de travail et qui est devenu une réelle industrie en expansion au cours des vingt cinq dernières années. Plusieurs compagnies explorent, et réalisent dans pas mal de cas, des produits qui incorporent des traitements des langues basés sur des grammaires. Plusieurs grammaires à large couverture ont été développées sur plusieurs années, voir des décennies, coordonnées de manière habituelle par un unique grammairien faisant souvent appel à des collaborateurs supplémentaires. L'une des plus grandes entraves au développement de ce type de grammaires est le coût trop élevé de leur construction. La LinGo Grammar Matrix vise à s'occuper de tout cela en fournissant un «starter-kit» qui tient compte d'un développement initial rapide et qui soutient des expansions à longs termes en représentant un ensemble d'hypothèses concernant des universaux trans-linguistiques. En gros, au lieu de concevoir une nouvelle grammaire de bout en bout, ou à partir d'une grammaire déjà existante mais à portée restreinte, la LinGo Grammar Matrix (disponible à partir de: http://www.delph-in.net/matrix/) permet de générer une «starter grammar» qui consiste au strict minimum d'entrées lexicales et de types de règles rudimentaires en configurant certaines propriétés typologiques et trans-linguistiques communes (cross-linguistically common typological properties) en accord avec la langue objet de l'étude.

4.3.2.3.2. La LinGO Grammar Matrix: de quoi s'agit-il exactement ?

Bien que l'ordre des mots et beaucoup d'autres phénomènes linguistiques varient à travers les langues, il y a toujours des modèles récurrents. Comme étape dans la conception d'une compatibilité maximale entre des grammaires de différentes langues — dans la mesure où les cadres théoriques le permettent — et pour permettre une réutilisation du code de grammaires à travers les langues, la HPSG Grammar Matrix (Bender et al., 2002) offre une structure de traits typée « racine» partagée, ayant déjà été utilisée par des grammaires existantes basées sur HPSG, et qui pourra être utilisée comme point de départ pour la construction de nouvelles grammaires pour différentes langues. Actuellement, cette structure «racine » est un extrait de la ERG (English Resource Grammar) (Copestake

& Flickinger, 2000), et offre donc des idées encodées dans cette grammaire à large couverture susceptibles d'être incorporées immédiatement dans d'autres grammaires. La Matrix est ainsi une tentative pour fournir une base fondée sur la typologie pour construire des grammaires des langues naturelles à partir d'un logiciel. Le noyau de la Matrix est un ensemble de types qui sont censés être universels. La version initiale de la Matrix inclut, entre autres, des types induisant la géométrie de base du signe linguistique, des types pour les types essentiels de règles lexicales (lexeme-to-word, lexeme-to-lexeme, word-to-word) et de règles syntaxiques (binary vs. unary, headed vs. non-headed, head final vs. head initial, head-complement, head-subject, head modifier, ...), des types induisant des représentations MRS, des types pour la manipulation de listes, ainsi que tous types ou hypothèses nécessaires à tout noyau de grammaire HPSG. De plus, et sans toutefois nier l'existence de phénomènes linguistiques très répandus mais non universels, la Matrix inclut des «libraries» qui consistent en des types additionnels couvrant plusieurs phénomènes non universels (Bender & Flickinger, 2005 ; Drellishak & Bender, 2005). La Matrix inclut aussi un système de paramétrage disponible en ligne et qui concerne la langue choisie, suggérant au linguiste un questionnaire pour la création d'une « starter grammar » basée sur la Matrix et les librairies appropriées et ajustée à la langue en question. La version actuelle du questionnaire inclut, entre autres, (i) des informations générales sur la langue (nom de la langue, Code ISO, …), des informations sur (ii) l'ordre des mots élémentaire des constituants majeurs dans les clauses matrices (SOV, SVO, VSO, V2, V-final, …), (iii) le nombre, (iv) la personne, (v) le genre, (vi) le cas, (vii) le temps, l'aspect et le mode (ceci permettrait directement de définir aussi bien des traits sémantiques liés au temps, à l'aspect et au mode, que des traits syntaxiques liés aux formes des verbes), (viii) l'existence de déterminants comme mots indépendants ainsi que l'ordre relatif entre le Nom et le Déterminant (Nom-Det vs. Det-Nom), (ix) les arguments NP vs. PP des verbes transitifs et intransitifs, (x) les stratégies pour l'expression de la négation de phrases matrices et des «yes/no questions», ainsi que les stratégies de la coordination des constituants, et bien d'autres encore. Le questionnaire permet en outre de définir des types lexicaux ainsi que des items lexicaux qui leur correspondent ; il permet aussi de créer des phrases de test. La section du lexique a été beaucoup améliorée pour permettre la description de morphologie flexionnelle complexe (O'Hara, 2008).

4.3.2.3.3. Avantages de la Matrix

La Matrix permet un saut considérable dans le développement de grammaires

à large couverture et à haute précision. Nous résumons les principaux avantages de l'adoption de cette grammaire comme point de départ dans le processus de développement de nouvelles grammaires électroniques comme suit:

- La Matrix est une open source. Elle peut donc être améliorée par n'importe quel développeur de grammaires.
- Elle peut être utilisée en analyse comme en génération.
- Son développement se fait dans un cadre de recherche très soutenue. Elle est écrite dans le cadre théorique des HPSG, elle utilise des représentations sémantiques en MRS et elle est développée sur la plate-forme LKB.
- Elle est utilisée dans un cadre trans-linguistique. Elle bénéficie donc des travaux dans beaucoup de langues.
- Son paramétrage a été simplifié par une interface web assez conviviale sous forme de questionnaire, qui fait appel à son tour à un script CGI (Common Gateway Interface) pour compiler la grammaire.
- Les grammaires compilées avec la Matrix sont certes encore de petites tailles, mais elles dépassent le cadre de grammaires jouets.
- Enfin, plusieurs langues utilisent la Matrix. Pour certaines, des grammaires à large couverture sont déjà disponibles (l'anglais et le japonais), pour d'autres il existe des grammaires assez consistantes (le norvégien, l'italien, l'espagnol et le grec) et pour d'autres encore, des travaux sont en cours.

5. Conclusion

Le processus de développement de grammaires électroniques nécessite trois phases. Une première phase consiste à déterminer les phénomènes de la langue à couvrir et leur dégager les analyses linguistiques adéquates. Les descriptions obtenues sont transposées, dans une seconde phase, dans un formalisme syntaxique. Enfin, une phase d'implantation où un environnement de développement est utilisé pour produire l'ensemble des composantes de la grammaire dont un lexique computationnel. Chacune de ces phases possède ses propres outils et son propre domaine d'investigation et donc une certaine autonomie mais les outils des trois phases pris ensemble doivent être compatibles et complémentaires pour accélérer le processus de développement de la grammaire. Combien même le développeur de grammaires électroniques ait une parfaite maitrise du processus et de ses outils, il est clair que ce processus consomme temps et hommes et nécessite par conséquent un travail collaboratif pour permettre une interaction rapide entre ses différentes phases. Cet article explique le processus de développement de grammaires électroniques à travers une certaine vision

de la grammaire et donc à l'aide d'un certain nombre d'outils qui ont fait leurs preuves dans le domaine du traitement automatique des langues naturelles d'une manière générale. La maitrise de ce processus permet au développeur de se fixer les objectifs et par conséquent la ou les phases à entreprendre ; rien n'empêche donc de faire en sorte qu'une phase soit déjà réalisée. Faut-il encore le rappeler, les outils ayant servis à l'écriture de cette revue ne sont que des exemples et sont donc propre à un cadre théorique particulier: les grammaires optant pour une approche à la fois lexicaliste et endocentrique. Le développeur est donc libre de choisir les outils qui lui conviennent. Dans notre présentation, nous avons ainsi délibérément choisit les grammaires d'unification mais çà aurait pu être tout autre type de grammaires pourvu que le cadre théorique choisit possède une implantation informatique. Un exemple de grammaires n'appartenant pas au courant des grammaires d'unification, ni même au courant générativiste transformationnel, sont les grammaires de dépendance dans leur version XDG (eXtensible Dependency Grammar) (Debusmann, 2006) pour lesquelles il existe une implémentation informatique: la XDK (eXtensible Dependency grammar Kit) (Debusmann & Duchier, 2005). Ensuite, dans le courant des grammaires d'unification, nous avons choisit la grammaire HPSG comme cadre théorique de modélisation et le système LKB comme plateforme d'implémentation de grammaires, et là encore ça aurait pu être la grammaire LFG (Kaplan et Bresnan, 1982) pour laquelle ont été écrites les deux plateformes: The Xerox LFG Grammar Writer's Workbench (Kaplan & Maxwell, 1996) et son successeur XLE (Xerox Linguistic Environment) (Butt, King, Nino et Segond, 1999). Mais ça aurait pu aussi être les grammaires TAG (Joshi et al., 1975) dont une implémentation est le système XTAG (Paroubek, Schabes et Joshi, 1992). Le processus en est toujours le même. Enfin, certes cet article se contente d'une revue d'un sous-ensemble des outils et des modèles de la littérature mais s'inscrit principalement dans une perspective applicative dans le sens où il présente un processus complet: des critères dits de Zwicky pour la description et l'analyse linguistique au système de performance LKB, en passant par le formalisme de modélisation des grammaires HPSG.

Notes

1- Les systèmes de performance peuvent être de nature humaine ou technologique. Parmi les systèmes de performance technologiques, il existe des systèmes dédiés à la modélisation de la compétence et de la performance, notions établit dans le cadre de la grammaire générative. Ceux qui nous intéressent ici ont été développés pour des grammaires à base de structures de trait (cf. infra). Il s'agit d'environnements de développement de grammaires, pouvant examiner la compétence et la performance des grammaires par le moyen de l'analyse et de la génération. Les (fragments de) grammaires intégrées à ces systèmes peuvent donner lieu à des traitements efficaces, imitant la rapidité du traitement des langues par les humains.

2- Notons que pour certains modèles linguistiques, les signes se composent de plusieurs structures contenant des informations phonétiques, syntaxiques, sémantiques et discursives. Ces informations linguistiques hétérogènes sont ramenées sous une notation commune en utilisant un formalisme tel que les structures de traits. Pour la sémantique par exemple, Sag et Wasow (1999) proposent un trait contenu qui permet d'accueillir a priori n'importe quelle théorie sémantique compositionnelle compatible avec les mécanismes de la grammaire syntagmatique (e.g. la Sémantique des Situations).

3- Il est à signaler que la représentation syntaxique d'une phrase par un arbre de dépendance est certainement plus ancienne que la représentation par un arbre syntagmatique. En effet, l'usage des dépendances remonte à l'antiquité. Les grammairiens arabes du 8ème siècle, comme Sibawayh, distinguaient déjà gouverneur et gouverné en syntaxe et utilisaient cette distinction pour formuler des règles d'ordre des mots ou de rection. On retrouve des représentations de structures de dépendance dans des grammaires du 19ème siècle. Mais la première théorie linguistique basée sur la dépendance est incontestablement celle de Tesnière (1934, 1959). La représentation syntaxique d'une phrase par une structure syntagmatique, quant à elle, ne s'est développée qu'à partir de Bloomfield (1933) et des travaux des distributionnalistes. (Owens, 1988 : 79–81, cité dans Kahane, 2001: 17)

4- Le syntagme nominal a connu deux grands types d'analyse: l'analyse NP (Noun Phrase analysis) et l'analyse DP (Determiner Phrase analysis). L'analyse en NP a été la première à voir le jour. L'idée de traiter les syntagmes nominaux comme des NP-s suppose que le substantif est l'élément «central» du syntagme nominal et que les déterminants, adjectifs, etc. sont des éléments

«périphériques». Par la suite, Abney (1987) proposa l'hypothèse DP. Selon cette hypothèse, l'analyse du syntagme nominal considère que le déterminant constitue la tête du syntagme nominal, en d'autres termes une projection du déterminant plutôt que le substantif.

5- En réalité, l'unification n'est qu'une opération parmi d'autres.

6- Un jeu de phrases de test est un outil utilisé pour l'évaluation d'analyseurs. Le jeu vise à couvrir les phénomènes syntaxiques majeurs pour la langue en se basant sur un vocabulaire très restreint. Il peut en outre ne concerner que les phénomènes considérés par l'évaluation. Le jeu contient à la fois des phrases grammaticales et des phrases agrammaticales.

7- Le système TRALE (Meurers, Penn et Richter, 2002 ; Penn, 2004 ; Muller, 2007), qui est un système concurrent de LKB, indique en plus pour chaque phrase le temps CPU en secondes que le système prend pour la traiter.

Bibliographie

- Abeillé, A. (2007). *Les grammaires d'unification*. Paris : Hermès Science publications: Lavoisier.
- Abeillé, A. & Blache P. (2000). «Grammaires et analyseurs syntaxiques». In J.-M. Pierrel (Ed.), *Ingénierie des Langues* (pp. 51–76). Paris: Hermès.
- Abney, S. P. (1987). *The English Noun Phrase in its Sentential Aspect*. PhD thesis, MIT, Cambridge, MA.
- Anderson, Stephen R. (1988). Morphological Theory. In F. Newmeyer (Ed.), *Linguistics: The Cambridge Survey, Vol (1*), 146–191. Cambridge: Cambridge University Press.
- Bender, E. M. & Flickinger, D. (2005). Rapid prototyping of scalable grammars: Towards modularity in extensions to a language-independent core. In Dale R., Wong K. F., Su J. and Kwong O. Y. (Eds.), *Proceedings of the 2nd International Joint Conference on Natural Language Processing IJCNLP-05 (Posters/Demos)* (pp. 203–8). Jeju Island, Korea.
- Bender, E.M., Flickinger, D.P. and Oepen, S. (2002). The grammar matrix: An open-source starter-kit for the rapid development of cross-linguistically consistent broad-coverage precision grammars. In Carroll, J., Oostdijk, N. and Sutcliffe, R. (Eds.), *Proceedings of the Workshop on Grammar Engineering and Evaluation at the 19th International Conference on Computational Linguistics* (pp. 8–14), Taipei, Taiwan.
- Blache P. (1999). *Contraintes et théories linguistiques : des Grammaires d'Unification aux Grammaires de Propriété*. Université Paris 7, Habilitation à diriger des recherches.
- Bloomfield, L. (1933). *Language*. London : Allen and Unwin.
- Butt M., King T. H., Nino M.-E. and Segond F. 1999. A Grammar Writer's Cookbook. *In CSLI Lecture Notes, N°95*. Stanford, CA : CSLI Publications.
- Chomsky N. (1957). *Syntactic structures*. La Haye : Mouton.
- Chomsky, N. (1970). Remarks on nominalization. In Jacobs, R. A. & Rosenbaum, P. S. (Eds.), *Readings in English Transformational Grammar, chapter (12)*, 184–221. Waltham, Massachusetts : Ginn and Company.
- Chomsky, N. (1981). *Lectures on Government and Binding*. Dordrecht : Foris Publications.
- Copestake A. & Flickinger D.P. (2000). An open source grammar development environment and broad coverage English grammar using HPSG. In

Proceedings of the 2nd International Conference on Language Resources and Evaluation. Athens, Greece.
- Copestake A. (2002). Implementing Typed Feature Structure Grammars. *In CSLI Lecture Notes, N°110*. Stanford, CA: CSLI Publications.
- Copestake, A., Flickinger, D., Pollard, C. and Sag, I. A. (2005). Minimal Recursion Semantics: An introduction. *Research on Language and Computation*, 3 (2–3), 281–332.
- Cori, M. (2003). La mathématisation des formalismes syntaxiques, *Linx* [En ligne], 48, pp.13–28, mis en ligne le 01 octobre 2003. URL : http://linx.revues.org/117 ; DOI: 10.4000/linx.117
- Debusmann R. (2006). *Extensible Dependency Grammar : A Modular Grammar Formalism Based On Multigraph Description*. PhD thesis, Saarland University.
- Debusmann R. & Duchier D. (2005). *Manual of the XDG Development Kit*.
- Drellishak, S. & Bender, E. M. (2005). A Coordination Module for a Cross-linguistic Grammar Resource. In Stefan Müller (Ed.), *The Proceedings of the 12th International Conference on Head-Driven Phrase Structure Grammar* (pp. 108–128). Stanford: CSLI Publications.
- Gaifman H. (1965). Dependency systems and phrase-structure systems. *Information and Control*, Vol (18), 304–337.
- Hays D. (1960). Grouping and dependency theories. Technical report RM-2646, Rand Corporation.
- Hays D. (1964). Dependency theory: A formalism and some observations. *Language*, 40(4), 511–525.
- Joshi, A. K., Levy L. S. and Takahashi M. (1975). Tree Adjunct Grammars. *Journal of Computer and Systems Sciences*, 10(1), 55–75.
- Joshi, A. K. (1987). An Introduction to Tree Adjoining Grammars. In Manaster-Ramer, A. (Ed.), *Mathematics of Language*. Amsterdam: John Benjamins.
- Kahane, S. (2001). Grammaires de dépendance formelles et Théorie Sens-Texte. In *Actes de TALN 2001, Vol (2)*, 17–76.
- Kaplan, R. M. & Maxwell, J. T. (1996). LFG grammar writer's workbench. Technical report, Xerox Corporation. ftp://ftp.parc.xerox.com/pub/lfg.
- Kaplan, R. M. & Bresnan, J. (1982). Lexical-Functional Grammar: A formal system for grammatical representation. In Joan Bresnan (Ed.), *The Mental Representation of Grammatical Relations* (pp. 173–281). Cambridge, MA : The MIT Press. Reprinted in Dalrymple, Kaplan, Maxwell and Zaenen (Eds.), *Formal Issues in Lexical-Functional Grammar* (pp. 29–130). Stanford: Center

for the Study of Language and Information. 1995.
- Meurers, W. D., Penn, G. and Richter, F. (2002). A Web-Based Instructional Plat-form for Constraint-Based Grammar Formalisms and Parsing. In Dragomir Radev & Chris Brew (Eds.), *Effective Tools and Methodologies for Teaching NLP and CL*, (pp. 18 – 25).

Müller, S. (2007). The Grammix CD Rom. A Software Collection for Developing Typed Feature Structure Grammars. In Tracy Holloway King & Emily M. Bender (Eds.), *Grammar Engineering across Frameworks 2007, Studies in Computational Linguistics ONLINE*. Stanford: CSLI Publications.
- O'Hara, K. (2008). *A Morphotactic Infrastructure for a Grammar Customization System*. MA Thesis, University of Washington.
- Oepen, S. (2001). [incr tsdb ()] – Competence and performance laboratory. Technical report, DFKI, Saarbrucken, Germany.
- Owens, J. (1988). *The Foundations of Grammar: An Introduction to Medieval Arabic Grammatical Theory*. Amsterdam: Benjamins.
- Paroubek, P., Schabes, Y. and Joshi, A. K. (1992). XTAG – A Graphical Workbench for Developing Tree-Adjoining Grammars. *4th Conference On Applied NLP* (pp. 223–7). Trento.
- Penn, G. (2004). Balancing Clarity and Efficiency in Typed Feature Logic Through Delaying. In *Proceedings of the 42nd Meeting of the Association for Computational Linguistics (ACL'04)*, Main Volume, 239–246. Barcelona, Spain.
- Pollard, C. & Sag, I.A. (1994). *Head-Driven Phrase Structure Grammar*. Chicago: University of Chicago Press.
- Pollard, C. J. & Sag, I. A. (1987). *Information-based Syntax and Semantics, Vol(1)*. Stanford: CSLI Publications [Distributed by University of Chicago Press].
- Sag, I. A., Bender, E. and Wasow, T. (2003). *Syntactic Theory: A Formal Introduction, 2nd Edition*. Stanford, CA: CSLI Publications.
- Sag, I. A. et Wasow, T. (1999). *Syntactic Theory: A Formal Introduction*. Stanford, CA: CSLI Lecture Notes, CSLI Publications.
- Shieber, S.M. (1990). Les grammaires basées sur l'unification. In Miller, P. & Torris, T. (Eds.), *Formalismes syntaxiques pour le traitement automatique du langage naturel* (pp. 27–85). Hermès.
- Tesnière, L. (1959). *Éléments de syntaxe structurale*. Paris: Librairie C. Klincksieck.
- The TRALE project: http://milca.sfs.uni-tuebingen.de/A4/Course/trale/

- The XLE project: http://www2.parc.com/isl/groups/nltt/xle/
- The XTAG project: http://www.cis.upenn.edu/~xtag
- van Eynde, F. (2006). NP-internal agreement and the structure of the noun phrase. *Journal of Linguistics, Vol(42)*, 139–186.
- Zwicky, A. M. (1977). *On Clitics*. Bloomington, IN: Indiana University Linguistic Club.
- Zwicky, A. M. (1985). Heads. *Journal of Linguistics, Vol(21)*, 1–29.