Grammaires de dépendances génératives probabilistes

Modèle théorique et application à un corpus français

Alexis Nasr

Laboratoire Lattice (CNRS - UMR8094)
UFRL, Université Paris 7 - Case 7003
2 Place Jussieu, 75251 PARIS cedex 05
alexis.nasr@linguist.jussieu.fr

RÉSUMÉ. *Nous décrivons dans cet article le formalisme grammatical des Grammaires de Dépendances Génératives Probabilistes (*GDGP*) ainsi que les résultats d'expériences d'analyse syntaxique réalisées à l'aide de grammaires extraites automatiquement à partir d'un corpus arboré français. Les* GDGP *se distinguent des grammaires probabilistes contemporaines tant par leur modèle algébrique que par leur modèle probabiliste. Le premier est un système génératif pour grammaires de dépendances et le second est un processus markovien permettant de conditionner l'établissement de dépendances d'un mot en fonction de certaines autres dépendances définies pour ce mot.*

ABSTRACT. *This paper describes the formalism of Probabilistic Generative Dependency Grammars (*GDGP*) as well as the results of parsing expriments with grammars that were automatically extracted from a french treebank.* GDGP *are different from contemporary probabilistic grammars with respect to both their algebraic and probabilistic models. The algebraic model is a generative system for dependency grammars and the probabilistic model is a Markov process which models the probability of a dependency between a governor and one of its dependent based on the other dependents of the gorvernor. The parser used for the experiments relies on a supertagger which selects, prior to parsing, the part of the grammar that will be used by the parser.*

MOTS-CLÉS : *grammaires de dépendances, grammaires probabilistes, étiquetage grammatical, analyse syntaxique automatique, extraction automatique de grammaires.*

KEYWORDS: *dependency grammars, probabilistic grammars, supertagging, parsing, automatically extracted grammars.*

1. Introduction

Nous présentons dans cet article un formalisme grammatical appelé grammaires de dépendances[1] génératives probabilistes (GDGP) ainsi que les résultats d'expériences d'analyse syntaxique réalisées à l'aide de grammaires écrites dans ce formalisme. Ces grammaires ont été construites automatiquement et leurs paramètres estimés à partir d'un corpus français étiqueté pour la syntaxe (Abeillé *et al.*, 2003), appelé corpus LE MONDE dans la suite de ce document. Des expériences analogues à celles décrites ici ont déjà été effectuées sur le corpus anglais du Penn TreeBank et ont abouti à des résultats d'analyse syntaxique du niveau de l'état de l'art (Nasr *et al.*, 2006). Les résultats obtenus sur le corpus LE MONDE sont malheureusement inférieurs à ceux obtenus sur le Penn TreeBank. Nous reviendrons sur ce point dans la conclusion.

Les GDGP se distinguent des approches contemporaines en analyse syntaxique probabiliste (en particulier (Magerman, 1995), (Eisner, 1996), (Charniak, 1997), (Collins, 1999) et (Collins, 2003)) par plusieurs aspects, portant aussi bien sur leur modèle algébrique que sur leur modèle probabiliste[2].

En ce qui concerne le modèle algébrique, contrairement aux approches citées ci-dessus, les GDGP ne reposent pas sur le formalisme des grammaires hors-contextes (CFG), ou des variantes de ce dernier, mais sur un modèle génératif pour grammaires de dépendances, appelé Grammaires de Dépendances Génératives (GDG), s'inscrivant dans la lignée des travaux de (Hays, 1964), (Gaifman, 1965) et (Abney, 1996). La raison de ce choix est que les travaux récents en analyse syntaxique probabiliste ont mis en avant la notion de dépendance syntaxique comme une notion clef pour la modélisation probabiliste de la syntaxe. C'est en effet l'établissement de dépendances entre deux mots de la phrase qui constitue l'événement élémentaire de ces modèles probabilistes. Dans les GDG, l'opération élémentaire de dérivation correspond précisément à l'établissement d'une dépendance et non à la décomposition d'un syntagme en une séquence d'autres syntagmes comme dans une CFG. Il y a ainsi correspondance entre les opérations élémentaires du modèle algébrique et les événements du modèle

1. Il est difficile de donner ici une définition générale des grammaires de dépendances. Cependant, en voici une caractérisation rapide : les structures de dépendances constituent un outil de description de la syntaxe des phrases, fondé sur la notion de dépendance entre des couples de mots de ces dernières. Un des mots du couple est appelé gouverneur, et l'autre dépendant. L'ensemble des dépendants possibles d'un gouverneur est appelé sa *valence* ou parfois, sa *valence active*. La structure syntaxique d'une phrase se résume à indiquer, pour chaque mot de la phrase, son gouverneur. L'ensemble des dépendances de la phrase constitue un arbre, appelé arbre de dépendances, dont les mots sont représentés par les nœuds de l'arbre et les dépendances, par ses branches. Pour une définition plus complète, on pourra se référer à (Tesnière, 1959) ou à (Mel'čuk, 1988).
2. On peut distinguer deux composantes dans une grammaire probabiliste. D'une part le *modèle algébrique*, qui est le formalisme servant de base à la grammaire probabiliste - par exemple les grammaires hors-contexte, les grammaires d'arbres adjoints ou, dans notre cas, les grammaires de dépendances génératives - et, d'autre part, le *modèle probabiliste* qui est l'ensemble des paramètres probabilistes définis par la grammaire, lesquels permettent d'attribuer une probabilité aux arbres et aux phrases générés.

probabiliste. D'un point de vue pratique, cette correspondance permet une meilleure articulation entre le modèle algébrique et le modèle probabiliste de la grammaire, comme nous le verrons dans la section 3.

En ce qui concerne le modèle probabiliste, la majorité des grammaires probabilistes actuelles repose sur le modèle *bilexical*. Les événements élémentaires de ce modèle correspondent à l'établissement de dépendances entre couples d'items lexicaux particuliers. On a longtemps cru que le bilexicalisme était à l'origine des progrès enregistrés par les analyseurs syntaxiques probabilistes actuels. Cependant, des études récentes, en particulier (Bikel, 2004), ont montré que les bons résultats de ces analyseurs ne provenaient pas des probabilités bilexicales, mais d'autres aspects, non lexicaux, des modèles probabilistes. Une caractéristique importante qui distingue les GDGP des approches contemporaines est qu'elles ne reposent pas sur le modèle bilexical. L'événement fondamental sur lequel repose leur modèle probabiliste est l'établissement d'une dépendance entre deux *catégories* de mots. La grammaire attribue donc une probabilité à une structure syntaxique de dépendances composée de catégories. Le nombre et la nature des ces dernières est variable ; ils dépendent, en particulier, du paramétrage du processus de construction de la grammaire.

L'attribution de catégories de la grammaires aux mots qui constituent la phrase à analyser est réalisée lors d'une étape dite d'étiquetage grammatical (*supertagging* en anglais) dont le principe a été proposé par Srinivas Bangalore et Aravind Joshi (Bangalore, 1997; Bangalore *et al.*, 1999) pour les grammaires d'arbres adjoints (TAG), mais qui peut être mis en œuvre pour tout type de grammaires lexicalisées (par exemple des grammaires catégorielles (Clark, 2002)). L'idée qui sous tend l'étiquetage grammatical est proche de celle de l'étiquetage morpho-syntaxique : il s'agit d'attribuer à tous les mots d'une phrase une étiquette dénotant une catégorie. Cependant, dans le cadre de l'étiquetage grammatical, du fait de la lexicalisation de la grammaire, les étiquettes correspondent à des objets décrivant les propriétés combinatoires d'un mot, en d'autres termes, les règles de la grammaire. Ainsi, à l'issue de la tâche d'étiquetage, une partie de la grammaire a été sélectionnée (l'ensemble des objets grammaticaux correspondant aux catégories sélectionnées par l'étiqueteur) et c'est à l'aide de cette seule partie que l'analyse syntaxique sera réalisée. C'est cette caractéristique de l'étiquetage grammatical qui fait dire à S. Bangalore et A. Joshi, qu'il s'agit « presque » d'analyse syntaxique (*Supertagging is almost parsing*). La raison pour laquelle nous avons eu recours à un étiqueteur grammatical est d'ordre pratique. En effet, la taille des grammaires utilisées ne permettait pas de réaliser l'analyse sans un filtrage préalable de la grammaire, réalisé, dans notre cas, par l'étiquetage grammatical.

Dans un système d'analyse qui distingue les deux étapes d'étiquetage grammatical et d'analyse syntaxique, les choix qui doivent être réalisés pour effectuer l'analyse syntaxique d'une phrase (sélection d'une catégorie pour un mot et établissement de dépendances entre couples de catégories) sont par conséquent réalisés par deux processus différents : l'étiqueteur et l'analyseur. La difficulté des deux tâches est fortement dépendante du nombre de catégories définies par une grammaire : un nombre réduit de catégories simplifie la tâche de l'étiqueteur et complexifie la tâche de l'analyseur. Il est ainsi possible, en modifiant le nombre de catégories définies par la grammaire d'influer sur la répartition du travail entre les deux processus, comme nous le verrons dans la section 5.

Le système d'analyse qui nous a permis de réaliser les expériences décrites dans la section 5 repose sur plusieurs processus de traitement, en particulier un module de construction de grammaires, un étiqueteur grammatical, un analyseur proprement dit et un module de recherche de l'analyse la plus probable parmi les analyses produites par l'analyseur. Nous avons fait le choix de privilégier dans cet article la description des formalismes de représentation des données et les résultats des expériences au détriment des processus. Le lecteur intéressé trouvera les détails concernant le processus de construction de grammaires dans (Chen, 2002; Dybro Johansen, 2004) et les détails concernant l'algorithme d'analyse et de recherche de l'analyse la plus probable dans (Nasr, 2004).

La structure de l'article est la suivante. Dans la section 2, nous décrivons les grammaires de dépendances génératives (GDG) qui constituent le modèle algébrique des GDGP. La section 3 décrit la partie probabiliste des GDGP, en particulier les principales hypothèses d'indépendance sur lesquelles repose ce modèle. Le processus d'extraction de grammaires GDGP à partir d'un corpus annoté syntaxiquement est brièvement décrit en 4 où l'on décrit aussi les résultats de l'extraction sur le corpus LE MONDE. La section 5 présente les résultats de l'analyse syntaxique à l'aide des grammaires décrites en 4. L'article s'achève sur la section 6, qui donne les conclusions auxquelles nous avons abouti et les directions qui se présentent pour poursuivre ce travail.

2. Les grammaires de dépendances génératives

Le formalisme des grammaires de dépendances génératives (GDG), décrit dans cette section, s'inscrit dans la lignée des travaux initiés par (Hays, 1964) et (Gaifman, 1965) visant à représenter une grammaire de dépendances à l'aide de (variantes de) CFG. L'idée maîtresse qui sous-tend ces travaux consiste à modifier la signification de la relation de dominance qui représente, dans les grammaires syntagmatiques, la relation d'inclusion de syntagmes au sein d'un autre syntagme, pour l'interpréter comme la relation de dépendance entre deux mots de la phrase. D'une manière schématique, une règle de la forme $C_1 \rightarrow C_2\, C_3$ indique qu'un mot de catégorie C_1 peut gouverner, dans une proposition, un mot de catégorie C_2 et un mot de catégorie C_3. Cette idée a été reprise ensuite par Steven Abney (Abney, 1996) qui a proposé de remplacer les parties droites de telles règles par des expressions régulières sur l'alphabet des catégories, permettant ainsi de représenter les dépendances répétables d'un même gouverneur. Le formalisme que nous proposons ici est très proche de celui proposé par S. Abney. Il s'en distingue néanmoins en représentant de façon explicite la notion de rôle fonctionnel, et en modélisant les règles génératives sous la forme d'automates finis d'un type particulier, appelés *automates lexicalisés*. Le remplacement des expressions régulières proposées par S. Abney par des automates finis ne modifie, bien entendu, pas les capacités génératives du formalisme. L'intérêt de de ce mode de représentation apparaîtra plus tard, en particulier en section 3, lors de l'introduction de probabilités dans les GDG. On verra que l'existence de plusieurs automates différents, reconnaissant un même langage, pourra être mis à profit pour mieux articuler le modèle probabiliste et le modèle algébrique d'une grammaire probabiliste.

2.1. *Définition*

Une GDG se présente comme un ensemble d'automates d'un type particulier, appelés *automates lexicalisés*. Un automate lexicalisé associé à un mot m décrit tous les dépendants possibles de m, en d'autres termes sa valence active ; m est appelé l'*ancre* de l'automate. Chaque automate possède un *nom*, qui définit une catégorie morpho-syntaxique. Cette dernière spécifie, outre la partie de discours de l'ancre de l'automate, la valence de cette dernière. Dans la suite de cet article, nous emploierons indifféremment les termes *automate* ou *catégorie* dans le contexte des GDG. Un exemple d'automate lexicalisé, de nom V, est représenté dans la partie supérieure de la figure 1. Un tel automate[3] indique que le verbe *mange* possède un dépendant sujet[4], obligatoire et non répétable, dont la catégorie est *nom* ou *pronom*, un dépendant objet qui est optionnel et non répétable, et des dépendants circonstanciels, optionnels et répétables introduits par une préposition. Les transitions de l'automate sont étiquetées par des couples $\langle f, c\rangle$, où f est un rôle fonctionnel et c une catégorie (le nom d'un automate), ou par des couples $\langle \texttt{LEX}, m\rangle$, où m est une ancre de l'automate. L'ordre des dépendants entre eux, dans la chaîne linéaire, et vis-à-vis de leur gouverneur est représenté par la structure de l'automate. Le dépendant le plus à gauche est le sujet ; il se trouve à gauche du gouverneur. L'objet direct éventuel se trouve à droite du gouverneur, et peut être séparé de lui par des compléments prépositionnels. D'autres automates lexicalisés sont représentés dans la figure 2.

Chaque mot (au sens de la théorie des langages) reconnu par un automate est une séquence de couples $\langle f, c\rangle$. Cette séquence correspond à un arbre de dépendances de profondeur 1, que l'on appellera *arbre élémentaire* de la grammaire. Trois exemples de tels arbres sont représentés dans la partie inférieure de la figure 1. Le mot correspondant à l'arbre de gauche est : $\langle \texttt{SUBJ}, N\rangle\ \langle \texttt{LEX}, \mathrm{mange}\rangle\ \langle \texttt{CIRC}, P\rangle$.

3. L'état initial d'un automate est identifié par une flèche pointant sur ce dernier. Ses états d'acceptation sont représentés en gras. Les transitions vides sont représentées en pointillés.

4. On remarquera que l'automate de la figure 2 sur-génère. Il ne contraint pas, en particulier, l'accord en nombre entre le sujet et le verbe. La prise en compte de telles contraintes au sein des grammaires GDG ne peut s'effectuer qu'au moyen de la multiplication des catégories. De telles catégories pourraient être représentées par des structures de traits, ce qui permettrait de prendre en compte certaines contraintes par unification des structures de traits, à l'image de ce qui est fait dans les grammaires d'unification.

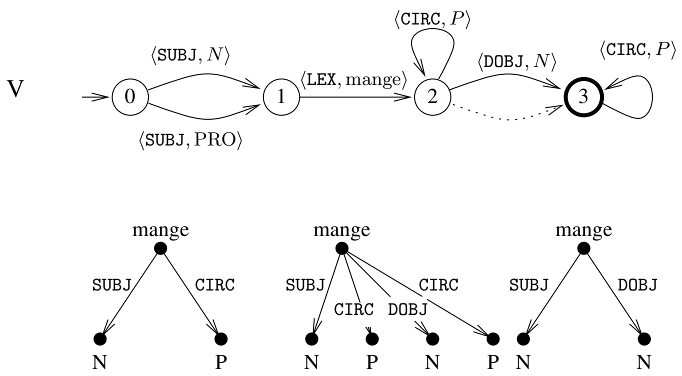


Figure 1. *Un automate lexicalisé et trois arbres élémentaires*

Le processus de dérivation à partir d'une GDG possède beaucoup de points communs avec la traversée d'un réseau récursif de transitions (Woods, 1970). Il fait appel, en particulier, à une pile. Cette dernière contient des couples $\langle c, e\rangle$ où c est le nom d'un automate de la grammaire et e est un état de c. Le recours à une pile s'explique par le fait que, durant la génération, plusieurs automates sont parcourus en parallèle. Ces derniers sont stockés dans la pile. L'opération élémentaire de dérivation consiste en la traversée d'une transition d'un automate. Ce franchissement correspond à l'établissement d'une dépendance entre les ancres de deux automates (sauf dans le cas de franchissement de transitions vides ou de transitions étiquetées$\langle \mathtt{LEX}, \mathrm{mot}\rangle$). Le résultat d'une dérivation est une séquence de transitions : les transitions franchies lors de la dérivation. Une telle séquence de transitions peut être associée de manière bi-univoque à un arbre de dépendances.

2.1.1. *Définition formelle*

D'un point de vue formel, une grammaire GDG est définie par un 6-tuplet $\langle \mathcal{C}, \Sigma, \mathcal{F}, \mathcal{A}, \theta, \mathcal{I}\rangle$ où $\mathcal{C}$, Σ et $\mathcal{F}$ sont des ensembles de symboles. $\mathcal{C}$ est l'ensemble des catégories, Σ est l'ensemble des éléments lexicaux, et $\mathcal{F}$ est l'ensemble des étiquettes fonctionnelles. $\mathcal{A}$ est un ensemble d'automates lexicalisés, décrits plus en détails ci-dessous. θ est une fonction bijective, qui associe à tout élément de $\mathcal{A}$ un élément de $\mathcal{C}$. C'est cette fonction qui définit une relation bi-univoque entre les automates et les catégories. $\mathcal{I}$ est un sous-ensemble de $\mathcal{C}$ appelé ensemble des symboles *initiaux*, lesquels jouent le rôle de l'axiome dans une grammaire générative : c'est par l'un de ces symboles que débute une dérivation.

Etant donné les trois ensembles de symboles $\mathcal{C}$, Σ et $\mathcal{F}$, un automate lexicalisé est un automate fini construit sur l'alphabet $(\mathcal{F} \times \mathcal{C}) \cup (\mathtt{LEX} \times \Sigma)$. En d'autres termes, chaque transition de l'automate est étiquetée par un couple $\langle f, c\rangle$ où f est une étiquette fonctionnelle et c une catégorie, ou par un couple $\langle \mathtt{LEX}, m\rangle$ où LEX est un symbole par-

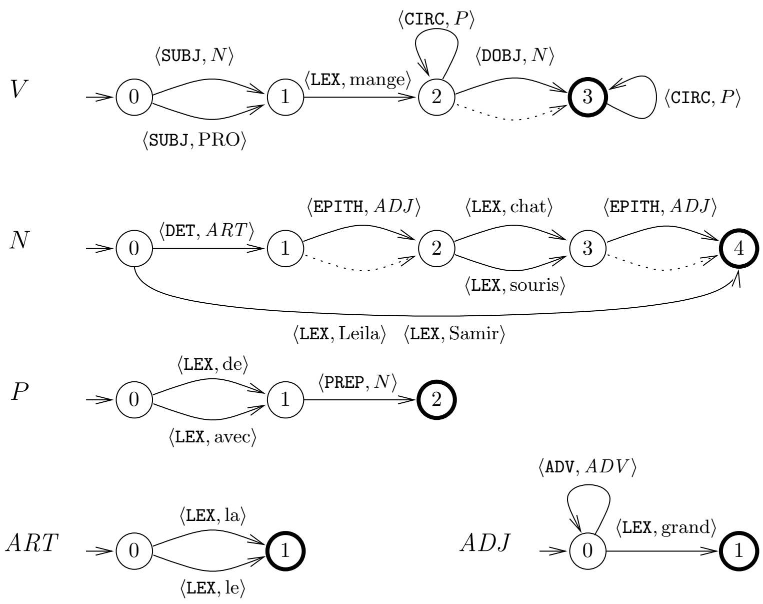


Figure 2. *Un exemple de grammaire* GDG

ticulier n'appartenant pas à l'ensemble des étiquettes fonctionnelles ($\mathcal{F}$), et m est un élément lexical ($m \in \Sigma$). Une telle transition est appelée *transition lexicale*. Tout chemin d'un automate lexicalisé menant de l'état initial à un état d'acceptation possède une transition lexicale et une seule. Les éléments lexicaux d'un automate lexicalisé sont appelés les *ancres* de l'automate. Lorsque l'on remplace par la chaîne vide la partie lexicale de toutes les transitions lexicales, on obtient un *schéma d'automate*.

2.1.2. *Sites d'un automate*

Les transitions des automates lexicalisés ne jouent pas toutes le même rôle. Certaines décrivent des dépendances que les ancres de l'automate peuvent établir, d'autres permettent de sélectionner une ancre de l'automate tandis que les transitions vides n'ont aucune influence sur les arbres que l'automate permet de générer. Nous allons établir une classification plus précise des différents types de transitions, et regrouper certaines transitions d'un automate au sein d'ensembles de transitions appelés *sites* de l'automate. Un automate lexicalisé se présente alors comme un ensemble de sites reliés entre eux par des transitions appelées *transitions inter-sites*, comme l'illustre l'automate de la figure 3. L'introduction de la notion de site permet de faire émerger une organisation interne des automates qui n'apparaît pas lorsque ces derniers sont vus comme des ensembles de transitions. L'émergence de ce niveau intermédiaire de structuration va permettre une description plus simple de certaines opérations sur les automates. On dira en particulier qu'un site est *pourvu* lorsque, lors de la traversée de l'automate, une des transitions du site est empruntée. D'autre part, la notion de site permettra de décrire plus simplement différents types d'automates qui seront définis par la suite.

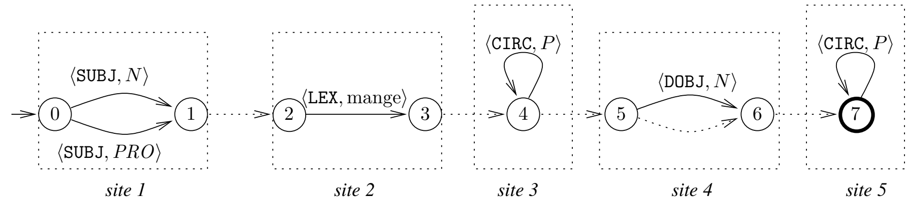


Figure 3. *Sites d'un automate*

Les transitions d'un automate qui décrivent des actants[5] des ancres de l'automate sont appelées *transitions actancielles*. Les transitions actancielles qui partagent la même étiquette fonctionnelle et le même état origine constituent un *site actanciel*, à l'image des transitions *sujet* et *objet* de l'automate de la figure 3, dont elles constituent les sites 1 et 4. Les sites actanciels se déclinent en deux types, les *sites obligatoires* et les *sites optionnels*. Les seconds se distinguent des premiers par la présence d'une transition vide permettant de ne pas pourvoir le site, à l'image du site 4 de l'automate de la figure 3.

Les transitions qui décrivent des modifieurs sont appelées *transitions modificatrices*. Elles sont regroupées au sein de *sites modifieurs*, comme les sites 3 et 5 de l'automate de la figure 3. Les sites modifieurs se déclinent eux aussi en deux types : des *sites répétables* et des *sites non répétables*. Les deux sites modifieurs de la figure 3 sont répétables, tandis que le site comprenant la dépendance *épithète* de l'automate N de la figure 2 offre un exemple de site non répétable. On remarquera que les sites modifieurs non répétables et les sites actanciels optionnels sont structurellement identiques (un faisceau de transitions entre deux mêmes états dont l'une est une transition vide), ils ne se distinguent que par la nature des étiquettes fonctionnelles de leurs transitions. Les transitions actantielles et modificatrices d'un automate sont appelées ses transitions *d'attachement*. Les sites actanciels et modifieurs constituent, eux, les *sites d'attachements* de l'automate.

Les transitions lexicales qui relient deux mêmes états forment un *site lexical*. Un automate peut avoir plusieurs sites lexicaux, à l'image de l'automate N de la figure 2.

Un automate lexicalisé dont les sites répétables sont composés de boucles sur un état unique est sous *forme canonique*. L'automate de la figure 3 offre un exemple d'au-

5. On distingue traditionnellement dans les grammaires de dépendances deux types de dépendants d'un gouverneur : ses actants et ses circonstants (dans la terminologie de (Tesnière, 1959)). Ils se distinguent par le rôle qu'ils jouent vis-à-vis du gouverneur. La limite entre les deux n'est pas facile à tracer et varie selon les auteurs. Nous avons gardé le terme actant mais avons inclu les circonstants dans la classe plus générale des modifieurs.

tomate sous forme canonique. On verra, dans la section 3, des automates lexicalisés qui ne sont pas sous forme canonique, ils s'en distinguent par la structure de leurs sites répétables.

2.2. *Comparaison avec d'autres approches*

L'utilisation d'automates finis pour la représentation de grammaires non régulières n'est pas une nouveauté. L'exemple le plus illustre est probablement le modèle des réseaux de transition récursifs (Recursive Transition Networks) de (Woods, 1970), formellement équivalent au modèle des grammaires hors-contexte.

D'autres travaux plus récents ont repris l'idée de représenter les règles d'une grammaire de dépendances sous la forme d'automates. Le modèle des *head automata* de (Alshawi, 1996) définit aussi des automates associés à des entrées lexicales, appelées têtes de l'automate. L'automate décrit aussi les différents dépendants de sa tête, plus précisément les catégories de ces derniers. La principale différence avec les automates lexicalisés se situe au niveau de la structure des deux types d'automates. Les automates lexicalisés génèrent les dépendants de ses ancres de gauche à droite alors que les *head automata* génèrent d'abord les dépendants les plus éloignés de la tête puis, au fur et à mesure de la génération, les dépendants s'en rapprochant. Les dépendants gauches et droits de la tête sont générés sur deux bandes d'écriture différentes, permettant ainsi d'entremêler génération de dépendants gauches et génération de dépendants droits. Si cette différence entre les deux types d'automates est la plus visible, elle n'est pas pour autant fondamentale et il est facile de transformer un automate lexicalisé en *head automaton* et vice-vesa. La différence la plus importante entre ces deux types d'automates apparaîtra plus clairement dans la section suivante. Nous verrons, en effet, que les automates lexicalisés peuvent adopter des structures différentes pour représenter des modèles probabilistes différents, alors que les *head automata* sont associés à un modèle probabiliste unique.

Le modèle des *automates bilexicaux* de (Eisner, 2000) constitue une autre approche récente reposant sur la notion d'automates. Ce modèle s'inscrit dans le paradigme des modèles bilexicaux que nous avons mentionné dans l'introduction. Les automates bilexicaux se distinguent des automates lexicaux des GDG et des *head automata* en ne manipulant pas des catégories mais uniquement des items lexicaux. Il s'agit là de la principale caractéristique de ce modèle. Un automate associé à l'item lexical m décrit tous les items lexicaux que m peut gouverner. À l'instar des *head automata*, les dépendants gauches et droits sont générés par deux automates différents, un automate gauche et un automate droit.

3. Grammaires de dépendances génératives probabilistes

Le processus de dérivation, dans le cadre d'une grammaire générative, est généralement non déterministe. À de nombreuses étapes de la dérivation (souvent à chacune d'entre elles), un choix doit être effectué parmi plusieurs issues possibles. Dans une grammaire générative probabiliste, une probabilité est associée à chacune de ces décisions. Le produit de ces probabilités constitue la probabilité de la dérivation, ou encore, de l'arbre produit. Plusieurs types de grammaires probabilistes peuvent être définies dans ce cadre, qui se distinguent par le modèle algébrique sur lequel elles reposent ou par le modèle probabiliste qu'elles définissent, en particulier sur les hypothèses d'indépendance de ce dernier. Nous commençons, en 3.1, par un rapide tour d'horizon des grammaires probabilistes contemporaines, avant de définir, en 3.2 les GDGP, qui ont les GDG pour modèle algébrique. Deux types de GDGP, implémentant des modèles probabilistes différents, sont décrits en 3.3. Nous terminons, en 3.4, par la définition de la probabilité d'un arbre et d'une phrase, étant donné une GDGP.

3.1. *Les grammaires probabilistes contemporaines*

Les CFG constituent le modèle algébrique d'une série récente de travaux sur les grammaires probabilistes ayant eu un grand retentissement dans le domaine du TAL. Nous désignerons ce courant sous le nom de *grammaires probabilistes contemporaines*. Il comprend en particulier les travaux de (Magerman, 1995), (Eisner, 1996), (Charniak, 1997), (Collins, 1999) et (Collins, 2003). Ces travaux s'inscrivent dans le cadre général des *grammaires fondées sur l'historique* proposé par (Black *et al.*, 1992). Dans ce cadre, les décisions à prendre aux différentes étapes d'une dérivation concernent le choix du symbole non terminal à réécrire, ainsi que la règle permettant de réécrire ce symbole[6]. Etant donné une CFG G d'axiome S et un mot m (dans le sens de la théorie des langages), la dérivation gauche d'un arbre T correspondant à m peut être représentée par une suite de n arbres syntagmatiques $T_1 \dots T_n$, obtenue par l'application successive des règles $R_1, \dots, R_{n-1}$ sur la feuille non terminale la plus à gauche de la frontière de l'arbre courant :

$$S = T_1 \overset{R_1}{\Longrightarrow} T_2 \overset{R_2}{\Longrightarrow} \dots \overset{R_{n-2}}{\Longrightarrow} T_{n-1} \overset{R_{n-1}}{\Longrightarrow} T_n = T \qquad [1]$$

La probabilité associée à cette dérivation et, par conséquent, à l'arbre T, est le produit des probabilités associées à la décision de réécrire le symbole le plus à gauche de T_i par la règle R_i, étant donné l'arbre T_i. On obtient donc :

$$P(T) = \prod_{i=1}^{n-1} P(R_i|T_i) \qquad [2]$$

6. En se restreignant au cas des dérivations gauches, le choix du symbole à réécrire est évacué car ce dernier est le symbole non terminal le plus à gauche dans la proto-phrase courante. La seule décision à prendre concerne, par conséquent, la règle qui servira à réécrire ce non terminal.

Le modèle décrit en (2) n'est pas réaliste, car la partie conditionnelle des probabilités qu'il met en jeu[7] devient, au fur et à mesure de la dérivation, de plus en plus importante. Elle est en effet constituée de l'intégralité de l'arbre généré à cette étape de la dérivation. La variété que peuvent présenter les historiques (les arbres générés) est telle, que l'estimation des probabilités $P(R_i|T_i)$, étant donné un corpus, est quasiment impossible pour une grammaire réaliste. En effet, la majorité des événements aura une probabilité nulle d'avoir été observée dans le corpus. C'est la raison pour laquelle les historiques sont regroupés en classes d'équivalence, lesquelles consistent à ne garder que certains éléments de ces historiques. En d'autres termes, la probabilité $P(R_i|T_i)$ sera modélisée par $P(R_i|E[T_i])$, où $E[T_i]$ est la classe d'équivalence de l'historique T_i. Divers types de CFG probabilistes peuvent être définis dans ce cadre. Ils se distinguent par la définition qu'ils donnent des classes d'équivalence des historiques ou, en d'autres termes, par les hypothèses d'indépendance sur lesquelles ils reposent. Ce sont ces hypothèses qui permettent d'ignorer certains événements de la partie conditionnelle, et d'aboutir à des modèles possédant moins de paramètres et qui sont, par conséquent, plus faciles à estimer.

On pourra remarquer que dans les grammaires fondées sur l'historique, le modèle probabiliste et le modèle algébrique (les CFG) ne reposent en général pas sur les mêmes hypothèses. Le modèle algébrique est hors-contexte : il suppose que la règle utilisée pour réécrire un symbole non terminal ne dépend que de ce dernier et pas de son contexte, alors que le modèle probabiliste conditionne, en général, l'application d'une règle au contexte du symbole non terminal.

Le modèle le plus simple pouvant être défini dans le cadre des grammaires fondées sur l'historique est celui des grammaires hors-contextes probabilistes (notées PCFG) introduites par (Booth, 1969). Les PCFG associent une probabilité à chaque règle d'une CFG. Avec la contrainte que les probabilités de toutes les règles possédant le même symbole pour partie gauche somment à 1. Dans un tel modèle, les probabilités sont associées directement aux règles, indépendamment d'un historique. On peut voir ce modèle comme le plus simple dans le paradigme des grammaires fondées sur l'historique, celui pour lequel le seul élément de l'historique pris en compte est le symbole non terminal à réécrire. Il s'agit en quelque sorte du modèle probabiliste le plus proche de l'esprit des grammaires hors-contexte dans la mesure où les probabilités sont aussi des probabilités hors-contexte. Les modèles algébriques et probabilistes reposent sur les mêmes hypothèses.

Les nombreuses hypothèses d'indépendance sur lesquelles reposent les PCFG en font un modèle probabiliste trop pauvre pour discriminer les analyses correctes dans l'ensemble des analyses que la grammaire associe à une phrase. Parmi ces hypothèses d'indépendance, deux sont régulièrement évoquées par les promoteurs de modèles alternatifs. Nous présenterons ces deux hypothèses dans les deux sous-sections suivantes.

---

7. Dans le contexte des processus stochastiques, les événements constituant la partie conditionnelle des probabilités sont souvent appelés « l'historique » de l'événement qu'ils conditionnent.

3.1.1. *L'hypothèse d'indépendance structurale des* PCFG

La première hypothèse mise en cause est l'hypothèse d'indépendance structurale. Celle-ci stipule, comme nous l'avons vu ci-dessus, que le choix d'une règle pour la réécriture d'un symbole est indépendant du contexte de ce dernier. Cette hypothèse est facilement mise en défaut, comme l'ont montré (Jurafsky *et al.*, 2000) ou (Resnik, 1992), en remarquant qu'en anglais, la probabilité qu'un GN se réalise comme un pronom dépend de façon cruciale de la fonction syntaxique qu'il occupe dans la phrase. En particulier, en position sujet, cette probabilité sera élevée alors qu'en position d'objet direct, elle sera faible. Dans une PCFG comportant les deux règles $GN \rightarrow PRO$ et $GN \rightarrow DET\ N$, chacune de ces dernières est associée à une probabilité qui est indépendante du contexte du GN. Il est par conséquent impossible de conditionner par le contexte du GN le fait que ce dernier se réécrive $DET\ N$ ou PRO, à moins de multiplier les catégories syntagmatiques, et, par conséquent, les règles de la grammaire.

Diverses solutions ont été apportées à ce problème par les grammaires probabilistes contemporaines. Elles consistent, en général, à choisir dans l'historique de la probabilité d'une règle un certain nombre d'éléments importants à prendre en compte pour l'estimation de sa probabilité. Ce choix peut être réalisé de manière manuelle par le concepteur du modèle, comme dans (Collins, 1999) ou (Black *et al.*, 1992), ou automatiquement, par des algorithmes de classification, comme dans (Magerman, 1995). Dans le modèle de (Collins, 1999), par exemple, la probabilité d'une règle $P \rightarrow HR_1R_2$ est conditionnée par la longueur de la chaîne dérivée à partir du non terminal R_1. Dans le modèle de (Magerman, 1995), l'application d'une règle $A \rightarrow \alpha$, par exemple, pourra dépendre du non terminal ayant introduit A à une étape précédente de la dérivation, si l'algorithme de classification considère que cet élément d'information est pertinent.

3.1.2. *L'hypothèse d'indépendance lexicale des* PCFG

La seconde hypothèse généralement mise en cause est l'hypothèse d'indépendance lexicale. Dans une CFG, l'introduction d'un mot dans une dérivation passe par la réécriture d'un symbole pré-terminal à l'aide d'une règle d'insertion lexicale (règle ayant un pré-terminal pour partie gauche et un élément lexical pour partie droite). Ainsi, dans le cas d'une phrase comportant un verbe et un syntagme prépositionnel, $\ldots penser\, à\ DET\ N\ \ldots$ par exemple, la génération de la préposition *à* est réalisée indépendamment de la réalisation du verbe *penser*. Il est par conséquent impossible de représenter dans le modèle la dépendance lexicale pouvant exister entre ces deux éléments.

La solution classique, pour la prise en compte de telles dépendances, dans les grammaires probabilistes contemporaines, passe par l'introduction, dans les règles de la grammaire, de la notion de *tête lexicale*. Une manière simple de prendre en compte cette dernière consiste à définir les symboles non terminaux de la grammaire comme des couples $\langle c, m\rangle$ où c est une catégorie syntagmatique, et m est la tête lexicale du syntagme. Ainsi, la grammaire distinguera par exemple, les non terminaux $\langle GV, penser\rangle$ et $\langle GV, manger\rangle$. Une règle de la forme :

$GV \rightarrow \; V \; GP$

donnera lieu, entre autres, à la règle :

$\langle GV, penser\rangle \rightarrow \; \langle V, penser\rangle \; \langle GP, \grave{a}\rangle$

La probabilité de cette dernière dépend donc de la nature des têtes lexicales des syntagmes V et GP. Cette solution, ou des variantes, ont été mises en œuvre dans les grammaires probabilistes contemporaines. On pourra remarquer que les événements élémentaires pris en compte dans ce genre d'approches sont précisément l'établissement de dépendances. La règle précédente correspond à l'établissement d'une dépendance entre le verbe *penser* et la préposition *à*. Sa probabilité correspond à celle d'établir la dépendance citée. La prise en compte de l'importance des relations entre mots, et la représentation explicite de ces relations dans les grammaires probabilistes contemporaines - pourtant fondées sur le modèle syntagmatique - a donné naissance à des modèles hybrides, curieux mélange de grammaires syntagmatiques et de grammaires de dépendances. Ces grammaires sont aussi appelées grammaires *bilexicales* du fait que leur modèle probabiliste prend en compte la nature lexicale du gouverneur et du dépendant.

Les approches bilexicales ont connu un succès important, et c'est sur elles que reposent, à ce jour, les meilleurs analyseurs syntaxiques probabilistes. Néanmoins, comme nous l'avons mentionné dans l'introduction, des études récentes (Bikel, 2004), (Gildea, 2001) ont montré que les probabilités bilexicales n'étaient pas à l'origine des performances de l'analyseur de Michael Collins (Collins, 1999) qui est l'un des analyseurs les plus performants à cette date[8]. La faible influence des probabilités bilexicales provient du fait que les événements qu'elles modélisent (des cooccurrences lexicales) sont assez rares pour que la majeure partie des cooccurrences apparaissant dans une phrase à analyser n'ait pas été observée dans un corpus d'apprentissage. Par conséquent, leur probabilité est nulle. Dans de tels cas, l'analyseur utilise d'autres variables, plus grossières, qui ne prennent pas en compte la nature lexicale des dépendants.

La situation pourrait être résumée de la façon suivante. D'une part, les grammaires ne prenant en compte que la partie du discours semblent trop grossières pour modéliser certaines distinctions. D'autre part, les grammaires bilexicales semblent trop fines. Lors de l'analyse, les événements qui se présentent à l'analyseur n'ont souvent pas été observés dans le corpus d'apprentissage. Nous verrons en section 3.2 une solution possible à ce problème.

Nous avons rappelé, en début de cette section, que la probabilité de générer un arbre étant donné une grammaire probabiliste, se décomposait en un produit de proba-

---

8. L'analyseur de M. Collins repose sur les PCFG, auxquelles il ajoute un grand nombre de dépendances, dont les dépendances lexicales ne sont qu'un aspect. C'est en enlevant les dépendances lexicales du modèle et en répétant les expériences de M. Collins que la faible influence des dépendances lexicales a été révélée par (Gildea, 2001).

bilités d'événements élémentaires, qui dépendaient de la nature du modèle algébrique sous-jacent. Dans le cas des PCFG, ces événements correspondent à la réécriture d'un symbole non terminal par une règle. Dans le cas des GDG probabilistes, le processus de génération, tel qu'il a été évoqué dans la section 2, se décompose en une séquence de franchissements de transitions des automates de la grammaire. Ces opérations de franchissement constituent les événements élémentaires sur lesquels vont être construits les modèles probabilistes. Ces derniers sont intimement liés à la structure des automates. Différents modèles probabilistes peuvent être définis, qui induiront des automates de structures différentes. Pour reprendre la distinction entre modèle algébrique et modèle probabiliste, on peut dire que dans le cas des GDGP, le modèle algébrique peut s'adapter à des modèles probabilistes différents, tout en restant dans le cadre des GDGP. La principale conséquence pratique de ce fait est que les algorithmes développés pour les GDGP, en particulier l'algorithme d'analyse syntaxique et l'algorithme de recherche de l'arbre le plus probable dans la forêt produite par l'analyseur peuvent être utilisés pour des grammaires implémentant des modèles probabilistes différents.

3.2. *Définition*

Une grammaire de dépendances générative probabiliste (GDGP) est une GDG dans laquelle les automates ont été enrichis de probabilités. Chaque transition t de chaque automate de la grammaire est ainsi associé à une probabilité, notée $p(t)$. De plus, chaque élément c_i de l'ensemble des symboles initiaux de la grammaire est associé à une probabilité, qui est notée $\pi(c_i)$ et appelée *probabilité initiale*.

Formellement, une GDGP est définie comme un 7-tuplet $\langle \mathcal{C}, \Sigma, \mathcal{F}, \mathcal{A}, \theta, \mathcal{I}, \pi \rangle$, dans lequel $\mathcal{C}$, Σ, $\mathcal{F}$ et $\mathcal{I}$ gardent la même définition que dans une GDG. $\mathcal{A}$ est un ensemble d'automates lexicalisés *probabilistes*, et π est la distribution de probabilités initiales, appelée elle-même *distribution initiale*.

Nous avons évoqué, en 3.1.2 et en 3.1.1, les deux hypothèses d'indépendance structurale et lexicale sur lesquelles reposaient les PCFG ainsi que les réponses qu'apportaient diverses grammaires probabilistes aux problèmes posés par de telles hypothèses. Nous allons voir dans les deux sous-sections suivantes les réponses apportées à ces problèmes par les GDGP.

3.2.1. *L'hypothèse d'indépendance structurale*

La principale hypothèse d'indépendance intrinsèque au modèle des GDGP est l'hypothèse markovienne, provenant de l'utilisation d'automates probabilistes. Cette hypothèse stipule que la probabilité d'une transition dépend de l'événement dénoté par l'étiquette de la transition, ainsi que de l'état duquel émane la transition, et pas du chemin ayant été suivi dans l'automate pour accéder à cet état. La probabilité d'une transition étiquetée c, émanant de l'état e_o et aboutissant à l'état e_d ($\langle e_o, c, e_d \rangle$) correspond à la probabilité conditionnelle $P(c|e_o)$. La correspondance entre le modèle probabiliste et le modèle algébrique s'effectue par l'intermédiaire de la structure des automates de la grammaire, plus précisément par la signification des états des automates. Cette dernière sera différente pour des modèles probabilistes différents.

Cette idée, bien connue, peut être illustrée simplement sur l'exemple de la figure 4. Les trois automates de cette figure définissent le même langage (a^*). Cependant, ils possèdent des structures différentes, et ils associent des significations différentes aux états. Chaque état représente les n derniers symboles ayant été générés dans le chemin menant à ce dernier. La valeur de n est différente pour les trois automates : 0 pour le premier, 1 pour le deuxième et 2 pour le troisième[9]. Les trois automates peuvent associer des probabilités différentes aux mêmes mots.

La probabilité du mot aaa, par exemple, vaudra dans le premier cas (automate de gauche) : $P(aaa) = P(a)^3$,

dans le deuxième cas : $P(aaa) = P(a) \times P(a|a)^2$

et dans le troisième : $P(aaa) = P(a) \times P(a|a) \times P(a|aa)$.

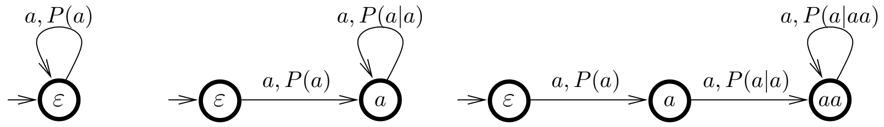


Figure 4. *Trois automates reconnaissant le même langage mais implémentant des modèles probabilistes différents*

On peut établir ici le lien entre le cadre général des grammaires fondées sur l'historique évoqué ci-dessus et les GDGP. Dans une GDGP, un événement correspond au franchissement d'une transition et l'historique est représenté par l'état duquel émane la transition. Cet historique dépend, comme nous l'avons vu ci-dessus, de la structure des automates, ou encore, des hypothèses d'indépendance du modèle probabiliste. Mais, quelles que soient ces dernières, l'historique est entièrement représenté dans l'automate : la probabilité d'un événement représenté par une transition d'un automate ne peut être conditionnée par un événement extérieur à ce dernier. En ce sens, les GDGP sont proches des PCFG : la probabilité d'une règle ne dépend pas d'événements extérieurs à la règle.

Nous verrons, en 3.3, deux types de GDGP qui définissent des automates de structures différentes, implémentant des modèles probabilistes différents. Un troisième modèle sera brièvement évoqué, il est décrit en détails dans (Nasr, 2004). En plus de l'hypothèse de Markov, propre aux cadre général des GDGP, ces deux types de grammaires imposent une hypothèse d'indépendance structurale plus contraignante, qui est l'*hypothèse d'indépendance des sites*. Cette dernière stipule que le choix d'une tran-

9. Nous avons choisi de donner aux états des noms correspondant aux n derniers symboles générés, afin que les probabilités $P(a|\textit{état})$ et $P(a|\textit{n derniers symboles})$ s'écrivent de la même manière.

sition dans un site est indépendant des choix effectués dans les autres sites de l'automate. Cette contrainte n'est pas intrinsèque aux GDGP. On peut envisager des GDGP ne la respectant pas.

3.2.2. *L'hypothèse d'indépendance lexicale*

Plusieurs réponses peuvent être apportées à la question de l'indépendance lexicale, selon le nombre d'ancres associées à un automate ; en d'autres termes, le nombre de transitions lexicales différentes qu'il comprend. Lorsqu'un automate comporte une seule ancre, alors les probabilités associées aux transitions de l'automate correspondent aux probabilités d'attachement de cet item lexical particulier. On se trouve alors dans le cas des probabilités bilexicales, du fait que chaque transition d'attachement correspond à l'établissement d'une dépendance entre deux items lexicaux. Comme nous l'avons évoqué ci-dessus, un tel modèle est confronté à de sévères problèmes d'estimation. En augmentant le nombre d'ancres par automate, on décide de ne plus distinguer ces items lexicaux du point de vue des probabilités des dépendances qu'ils peuvent établir. On diminue ce faisant le nombre de catégories et, donc, le nombre d'événements différents distingués par le modèle, en d'autres termes, son nombre de paramètres. En poussant à l'extrême ce processus de diminution du nombre de catégories, on peut aboutir à un seul automate (une seule catégorie) qui agrège toutes les règles de la grammaire. Nous adopterons une position intermédiaire, qui consiste à distinguer des catégories à partir de critères syntaxiques, en particulier leur valence active. Ainsi, tous les éléments lexicaux partageant la même valence active définissent une catégorie et le modèle probabiliste ne les distinguera pas les uns des autres : les probabilités seront définies sur l'espace des catégories.

3.3. *Types de GDGP*

Les deux types de GDGP définis ci-dessous respectent l'hypothèse d'indépendance des sites évoquée dans la section précédente. Ils se distinguent par la structure interne des sites, plus précisément des sites répétables. Le premier modèle décrit, appelé modèle initial et noté GDGP-INIT, définit des automates dont la structure est celle des automates canoniques. Le modèle suivant, appelé modèle bigramme (GDGP-2G) se distingue de GDGP-INIT en proposant des sites répétables plus complexes, permettant de modéliser plus finement le phénomène de l'attachement multiple à un site, principale source d'ambiguïté. Son nom provient du fait que la probabilité d'un attachement à un site répétable est conditionnée par le rattachement qui le précède, lorsque ce dernier existe. Un troisième modèle, appelé modèle positionnel (GDGP-POS) dont on ne pourra parler ici, faute de place, propose une autre modélisation probabiliste du phénomène de rattachement multiple : la probabilité d'un attachement est conditionnée par l'ordre de ce dernier (le premier attachement est d'ordre 1, le second d'ordre 2 etc). On trouvera dans (Nasr, 2004) une description complète de ce modèle.

3.3.1. *Le modèle initial :* GDGP-INIT

Comme nous l'avons vu en 2.1.2, un automate canonique se présente sous la forme d'une séquence de sites reliés entre eux par des transitions inter-sites. Les sites lexicaux, les sites obligatoires et les sites optionnels se présentent comme des faisceaux de transitions reliant deux états. Les sites répétables se présentent, eux, comme des ensembles de transitions modificatrices bouclant sur un même état. C'est cette caractéristique qui permet de modéliser des dépendances répétables. Les différents types de transitions d'un automate correspondent à des événements de nature différente, dont la probabilité est représentée par la probabilité de la transition.

Les probabilités d'attachement s'interprètent assez facilement. La probabilité d'une transition étiquetée $\langle f, c_2 \rangle$ sur un site i de l'automate c_1 correspond à la probabilité de l'attachement de l'automate c_2 sur le site i de l'automate c_1.

Les probabilités des transitions d'un site obligatoire correspondent à la probabilité de choisir un automate donné pour pourvoir le site. Chaque site obligatoire devant être pourvu par un automate et un seul, la somme des probabilités des transitions du site est égale à 1. L'intégralité de la masse des probabilités est par conséquent répartie sur les transitions du site. Lorsque le site est optionnel, alors ce dernier comporte une transition vide. La probabilité de cette transition est celle que le site ne soit pas pourvu.

Dans le cas d'un site répétable, la situation est plus délicate, dans la mesure où, contrairement aux autres sites, un site répétable peut décrire plusieurs attachements successifs. On ne sait à l'avance combien d'attachements différents s'effectueront sur ce site lors d'une dérivation, ni, par conséquent, sur quel nombre d'événements répartir la masse de probabilité. Lorsque plusieurs attachements d'un même automate se produisent sur un même site, ils possèdent tous la même probabilité. Intuitivement, cette modélisation semble bien pauvre, car la probabilité d'occurrence d'un premier attachement à un site devrait être différente de la probabilité de deuxième occurrence ...

La probabilité associée à une transition lexicale étiquetée $\langle \texttt{LEX}, m \rangle$, dans un automate c correspond à la probabilité conditionnelle $P(m|c)$. En d'autres termes, c'est la probabilité de choisir le mot m comme ancre de l'automate c. Tout chemin de l'automate passant par une transition lexicale et une seule, la somme des probabilités lexicales de l'automate est égale à 1. Dans le cas des schémas d'automates, les sites lexicaux sont réduits à une seule transition et la masse de probabilité est répartie uniformément entre les différents sites lexicaux de l'automate.

Les probabilités de transition décrivent la probabilité de passer d'un site à un autre. Lorsque les automates sont sous forme canonique, ces transitions ne présentent pas d'intérêt, dans la mesure où les automates se présentent comme une suite linéaire de sites. D'un site donné, on ne peut aller, au plus, que vers un seul autre site. La probabilité de chacune de ces transitions est par conséquent égale à 1.

3.3.2. *Le modèle bigramme :* GDGP-2G

Le modèle GDGP-2G se distingue de GDGP-INIT par la structure des sites modifieurs. En effet GDGP-INIT considérait que plusieurs rattachements à un même site répétable constituait des événements indépendants. Dans le modèle GDGP-2G, la probabilité d'attachement à un site répétable dépend de la nature de l'attachement précédent sur le même site. Nous avons représenté dans la partie droite de la figure 5 un site répétable bigramme et dans la partie gauche un site répétable GDGP-INIT permettant les mêmes attachements.

Formellement, les sites répétables GDGP-2G correspondent à des chaînes de Markov d'ordre 1, représentées sous la forme d'automates pondérés (Nasr *et al.*, 1999), (Allauzen *et al.*, 2003). Dans l'automate de la figure 5, l'état 1 (respectivement 2) de l'automate correspond au fait que l'attachement précédent soit l'automate c_1 (respectivement c_2). La transition reliant l'état 0 à l'état 1 (respectivement 2) correspond à la probabilité que le premier attachement à ce site soit l'automate c_1 (respectivement c_2). Cette probabilité s'écrit $P(c_1|\texttt{DEBUT})$ (respectivement $P(c_2|\texttt{DEBUT})$). Enfin, la transition vide menant de l'état 1 (respectivement 2) vers l'état 3 correspond au fait que le dernier attachement sur le site soit l'automate c_1 (respectivement c_2). Cette probabilité s'écrit $P(\texttt{FIN}|c_1)$ (respectivement $P(\texttt{FIN}|c_2)$). Le modèle bigramme peut être étendu à un modèle d'ordre supérieur, au prix d'une augmentation de la complexité des automates et du nombre de paramètres à estimer.

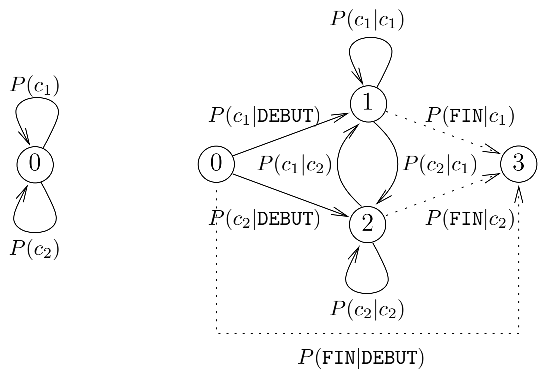


Figure 5. *Sites répétables d'une grammaire* GDGP-INIT *et d'une* GDGP-2G

On a représenté dans la figure 5 un site répétable GDGP-INIT (à gauche) et un site répétable GDGP-2G (à droite) reconnaissant le même langage.

L'automate de gauche associe au mot $c_1\ c_2\ c_1$ la probabilité :

$$P(c_1\ c_2\ c_1) = P(c_1)^2 \times P(c_2)$$

Tandis que l'automate de droite lui associe la probabilité suivante :

$$P(c_1\ c_2\ c_1) = P(c_1|\texttt{DEBUT}) \times P(c_2|c_1) \times P(c_1|c_2) \times P(\texttt{FIN}|c_1)$$

3.4. *Probabilité d'un arbre et probabilité d'une phrase*

Comme dans toutes les grammaires génératives probabilistes, la probabilité d'un arbre T, étant donné une GDGP $G = \langle \mathcal{C}, \Sigma, \mathcal{F}, \mathcal{A}, \theta, \mathcal{I}, \pi \rangle$ est le produit des probabilités des choix effectués lors de la génération de T. Le premier choix porte sur la sélection d'un élément de $\mathcal{I}$ comme premier automate de la dérivation, puis chacun des autres choix correspond au franchissement d'une transition dans un automate de la grammaire, lors du processus évoqué dans la section 2. La probabilité d'un arbre T, correspondant à la séquence de transitions $t_{1,n}$, est représentée en (3)[10].

$$P(T = t_{1,n}) = \pi(Auto(t_1)) \times P(t_{1,n}) = \pi(Auto(t_1)) \times \prod_{i=1\ldots n} P(t_i) \qquad [3]$$

L'équation (3) est valable pour les deux types de GDGP vus ci-dessus et, de façon plus générale, pour tout type de GDGP.

La probabilité associée à une phrase S par une grammaire générative probabiliste G est définie de façon standard, comme la somme des probabilités des différents arbres syntaxiques que G permet d'associer à S, notés $T_G(S)$. La probabilité jointe $P(T, S)$ peut se réécrire grâce à la règle des probabilités conditionelles. Elle peut ensuite être simplifiée en remarquant que les arbres produits par le processus génératif sont ordonnés et que, par conséquent, une seule suite de mots correspond à un arbre. Par conséquent, $P(S|T) = 1$. Ces différentes étapes sont représentées dans l'équation 4.

$$P(S) \stackrel{def}{=} \sum_T P(T, S) = \sum_T P(S|T) \times P(T) = \sum_{T \in T_G(S)} P(T) \qquad [4]$$

4. Construction automatique de GDGP

Le processus de construction de GDG à partir d'un corpus O est réalisé de manière indirecte. Dans un premier temps, une grammaire au format des grammaires d'insertion d'arbres TIG est produite. Cette étape est réalisée à l'aide de l'algorithme proposé par John Chen (Chen, 2002), adapté au français par (Dybro Johansen, 2004). À l'issue de ce processus d'extraction, un 4-tuplet $\langle G, O', \mathcal{T}_A, \mathcal{T}_R \rangle$ est produit, où :

– G est la grammaire TIG, qui se présente sous la forme d'un ensemble de schémas d'arbres élémentaires, où chaque arbre élémentaire est associé à une catégorie ;

10. Etant donné une transition t, on note $Auto(t)$ l'automate auquel elle appartient.

– O' est un appariement qui associe, à toute occurrence d'un mot du corpus O, l'identifiant de l'arbre élémentaire de la grammaire G que cette occurrence ancre. O' peut aussi être vu comme un corpus étiqueté qui associe une catégorie à chaque occurrence d'un mot. C'est à partir de ce corpus que seront estimés les paramètres d'un étiqueteur syntaxique ;

– $\mathcal{T}_A$ est la table d'attachement, qui spécifie pour chaque arbre élémentaire T_i le nombre d'occurrences des différents attachements (insertion ou adjonction) observés sur les différents nœuds de T_i. Dans le cas d'une adjonction, est indiquée, en plus, l'adjonction précédente sur le même nœud. C'est à partir de ces comptes que sont estimées les probabilités des transitions d'attachement ;

– $\mathcal{T}_R$ est la table des racines. Elle indique, pour chaque schéma d'arbre élémentaire de G, le nombre de fois que ce dernier constitue la racine d'un arbre de dérivation dans le corpus O. C'est à partir de ces comptes que seront estimées les probabilités initiales de la grammaire.

La construction d'une grammaire GDGP $G' = \langle \mathcal{C}, \Sigma, \mathcal{F}, \mathcal{A}, \theta, \mathcal{I}, \pi \rangle$ à partir d'un 4-tuplet $\langle G, O', \mathcal{T}_A, \mathcal{T}_R \rangle$ se décompose en deux parties : une partie *algébrique* et une partie *probabiliste*.

La partie algébrique consiste à construire une GDG à partir de la TIG G. À l'issue de cette étape, un automate de l'ensemble $\mathcal{A}$ est construit pour tout arbre élémentaire de G. Les noms des automates, ou encore des catégories de la GDGP, sont les mêmes que les noms des schémas d'arbres, ou *supertag*s.

La partie probabiliste consiste à estimer les paramètres d'une GDGP, à partir des comptes contenus dans les tables $\mathcal{T}_A$ et $\mathcal{T}_R$.

Nous ne pourrons rentrer ici dans tous les détails de la production des grammaires. Nous commencerons par décrire très brièvement, en 4.1, le formalisme des TIG, moins connu que les TAG, dont il constitue un sous-ensemble. Nous donnerons ensuite, en 4.2, les principes de l'algorithme d'extraction des TIG puis, en 4.3 ceux de la construction de GDGP à partir de TIG. L'estimation des paramètres probabilistes est décrite en 4.4. Enfin, l'application de ces traitements sur le corpus LE MONDE sera décrit en 4.5.

4.1. *Les grammaires d'insertion d'arbres*

Les TIG ont été définies dans (Schabes *et al.*, 1995). À l'instar des TAG, les TIG définissent des arbres élémentaires initiaux et auxiliaires, et deux opérations de réécriture d'arbres : l'adjonction et la substitution. Cependant, les TIG introduisent des restrictions sur les arbres élémentaires, dont le résultat est de réduire la puissance générative du formalisme : on passe d'un système permettant de reconnaître des langages faiblement dépendants du contexte (Joshi *et al.*, 1975) à un système ne permettant de reconnaître que les langages hors-contexte. La contrepartie de cette perte de pouvoir génératif est l'existence d'algorithmes d'analyse, fondés sur la programmation dynamique, en $O(n^3)$.

Les restrictions qui distinguent les TIG des TAG portent sur les formes que peuvent prendre les arbres élémentaires auxiliaires. En effet, dans une TIG, les arbres auxiliaires sont tels que toutes les feuilles non vides se trouvent à gauche ou à droite du nœud pied. On parle alors d'arbres auxiliaires gauches et d'arbres auxiliaires droits. Les TIG proscrivent donc les arbres auxiliaires enveloppants (pour lesquels le nœud pied peut avoir des nœuds feuilles à sa gauche et à sa droite). Ce sont précisément ces derniers qui confèrent aux TAG une puissance générative supérieure à celui des CFG. Les TIG se distinguent néanmoins des CFG, en ce qu'elles lexicalisent fortement (selon la définition de (Schabes, 1990)) ces dernières.

4.2. *L'algorithme d'extraction des* TIG

Notre objectif n'est pas de donner ici une description détaillée de la procédure d'extraction d'une TIG développée par J. Chen. Il consiste simplement à offrir une idée générale de son fonctionnement ainsi que des connaissances linguistiques qui permettent de paramétrer le processus d'extraction et, par conséquent, les grammaires extraites. On trouvera dans (Chen, 2002) une description détaillée de ces travaux ainsi que les résultats de l'extraction sur le corpus Penn Treebank. L'adaptation de ce travail au corpus LE MONDE est décrit en détail dans (Dybro Johansen, 2004).

Etant donné un corpus arboré, l'algorithme d'extraction de grammaires prend en entrée une partie de ce dernier, appelée corpus d'apprentissage, et produit une TIG. Le corpus est traité phrase par phrase, arbre par arbre pour être plus précis. Pour chaque arbre syntagmatique T, en entrée, l'algorithme produit un arbre de dérivation TIG : des arbres élémentaires ainsi que la description des opérations d'adjonction ou d'insertion qui permettent de les combiner pour produire l'arbre T. Les arbres élémentaires sont créés en « découpant » T, de manière à former des arbres élémentaires initiaux ou auxiliaires. A l'issue de la création d'un arbre élémentaire, deux situations peuvent se présenter, selon que le schéma de cet arbre (l'arbre duquel on a éliminé l'ancre lexicale) a déjà été créé lors du découpage de T, ou d'un arbre précédent du corpus d'apprentissage, ou qu'il s'agit de la première occurrence d'un tel schéma d'arbre. Dans ce dernier cas, ce nouveau schéma est ajouté à la grammaire en cours de construction et un identifiant lui est associé. Cet identifiant correspond à une catégorie GDG.

La taille de la grammaire, calculée en nombre de schémas d'arbres élémentaires, croît au fur et à mesure du traitement du corpus. L'évolution de la taille de la grammaire en fonction de la taille du corpus est un aspect important du processus d'extraction. En effet, l'idée même d'une grammaire (ensemble fini de règles permettant de générer un ensemble infini de structures) suppose que le processus d'extraction de la grammaire converge : à partir d'une certaine taille de corpus, la taille de la grammaire ne doit plus croître.

La décomposition d'un arbre syntagmatique en arbres élémentaires repose d'une part sur un principe de découpage, et d'autre part sur des paramètres linguistiques qui vont guider le processus. Ces paramètres se présentent sous deux formes. La première est une table d'identification des têtes lexicales, ou *table de percolation* dont le principe a été introduit par (Magerman, 1995). Cette table regroupe un certain nombre d'heuristiques qui permettent d'associer une tête lexicale à chaque nœud syntagmatique d'un arbre. Cette association est réalisée en identifiant, grâce à la table de percolation, parmi les fils d'un nœud syntagmatique n, un *nœud tête* n_H, indiquant que la tête lexicale de n est la tête lexicale de n_H. Cette dernière est elle-même la tête lexicale du nœud tête de n_H, et ainsi de suite, jusqu'à atteindre les feuilles de l'arbre. Cette information permet de faire « remonter » les étiquettes des feuilles lexicales dans les nœuds internes de l'arbre syntagmatique. Un exemple du produit d'un tel processus est représenté dans la figure 6, extraite de (Dybro Johansen, 2004).

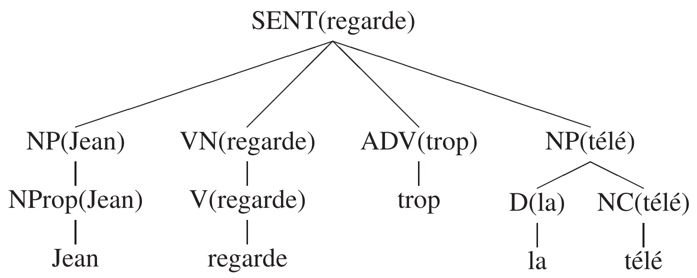


Figure 6. *Remontée des étiquettes lexicales dans un arbre syntagmatique*

L'autre information nécessaire au processus de découpage est la répartition des dépendants des mots de la phrase en actants et en modifieurs. Cette distinction est cruciale pour décider si un nœud de l'arbre syntagmatique en cours de découpage donnera naissance à un arbre auxiliaire ou à un arbre initial. La situation est représentée schématiquement dans la figure 7. Quatre des cinq fils du nœud syntagmatique étiqueté X de l'arbre de la partie gauche de la figure ont été étiquetés A_i ou C_i, selon qu'ils représentent un actant ou un modifieur de la tête de X. Cette dernière est identifiée, grâce à la table de percolation, comme étant la tête du fils étiqueté H. À l'issue du processus de découpage, les fils A_i donneront naissance à des nœuds d'insertion (les nœuds $A_1\downarrow$ et $A_2\downarrow$ de l'arbre α_1) et à des arbres initiaux (α_2 et α_3) tandis que les fils C_i donneront naissance à des arbres auxiliaires (β_1 et β_2).

4.3. *Transformation d'une* TIG *en une* GDG

La transformation d'une TIG G en une GDG G' sous forme canonique consiste à construire un automate lexicalisé pour chaque arbre élémentaire de G. La construction de l'automate A, correspondant à l'arbre élémentaire T, est réalisée par un parcours de ce dernier. Pour chaque nœud parcouru de T, toutes les opérations de substitution ou d'adjonction qu'il est possible d'effectuer à ce nœud sont envisagées. Pour chacune d'entre elles, une transition est construite dans A. Ces transitions constituent un site de l'automate. A représente ainsi toutes les opérations d'attachement potentielles pou-

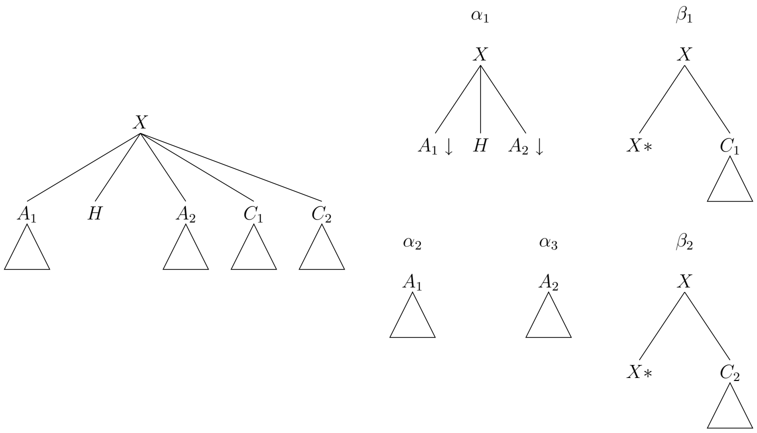


Figure 7. *Découpage d'un arbre dérivé en arbres élémentaires*

vant être réalisées sur T. La GDG produite distingue deux types de dépendances : les actants, étiquetés par le symbole `A`, et les modifieurs, étiquetés `C`. Le sens de parcours de T détermine l'ordre dans lequel les transitions apparaîtront dans l'automate. Ce dernier décrivant les dépendants d'un élément lexical de gauche à droite, le parcours de l'arbre doit s'effectuer de manière à respecter cet ordre. Pour cela, les arbres sont parcourus de gauche à droite, en profondeur d'abord. Lors du parcours d'un arbre, chaque nœud est visité deux fois : une fois à la descente, et une fois à la montée. Nous avons représenté, dans la partie gauche de la figure 8, un arbre élémentaire TAG dans lequel le parcours effectué pour construire un automate lexicalisé est matérialisé par des flèches en pointillés. L'automate construit est représenté dans la partie droite de la figure.

4.4. *Estimation des probabilités d'une GDG-2G*

Les différentes probabilités associées aux transitions d'une GDGP-2G sont estimées à partir des comptes collectés dans les tables $\mathcal{T}_R$ et $\mathcal{T}_A$ à l'issue du processus d'extraction de la grammaire. Nous détaillons ci-dessous l'estimation des différents types de probabilités.

Probabilités initiales

Les probabilités initiales π de G' sont estimées à partir de la table des racines $\mathcal{T}_R$ au moyen d'une simple estimation par maximum de vraisemblance. Pour tout automate A, la probabilité $\pi(A)$ n'est rien d'autre que le nombre de fois où l'automate A constitue la racine d'un arbre du corpus O, divisé par le nombre d'arbres composant le corpus.

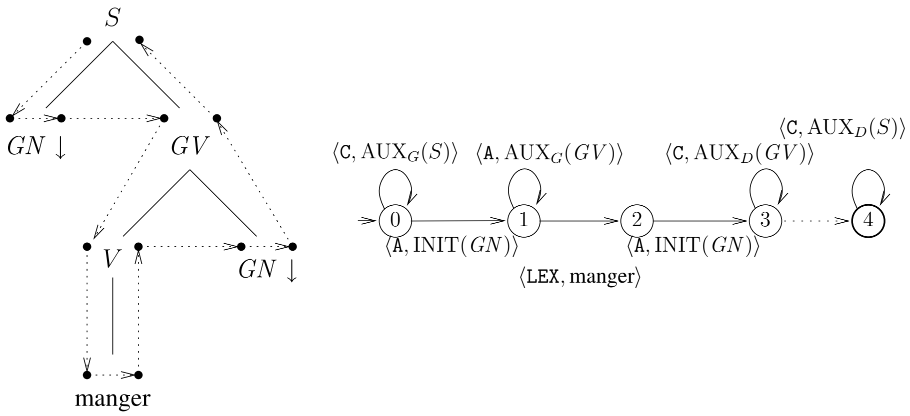


Figure 8. *Transformation d'un arbre élémentaire* TIG *en un automate* GDG. $\mathrm{AUX}_G(X)$ *(respectivement* $\mathrm{AUX}_D(X)$*) correspond à l'ensemble de tous les arbres auxiliaires gauches (respectivement droits) de la grammaire, ayant le non terminal X pour racine.* $\mathrm{INIT}(X)$ *correspond à l'ensemble de tous les arbres initiaux ayant le non terminal X pour racine*

Probabilités des transitions actancielles

Les probabilités associées aux transitions actancielles correspondent à la probabilité de choisir une catégorie donnée pour pourvoir un site actanciel. Elles sont calculées à partir des comptes contenus dans la table $\mathcal{T}_A$. Certaines transitions actancielles construites correspondent à des attachements qui n'ont jamais été observés dans le corpus d'apprentissage. Les comptes leur correspondant dans la table $\mathcal{T}_A$ sont par conséquent nuls. Afin d'attribuer une probabilité à ces événements, nous avons appliqué une méthode simple de lissage de probabilité, connue sous le nom de *add one smoothing*. Elle consiste à augmenter de 1 les comptes de tous les attachements. Ainsi, les événements jamais observés auront un compte de 1 et les événements observés n fois auront un compte de $n + 1$. Les probabilités sont ensuite calculées par maximum de vraisemblance sur ces nouveaux comptes.

Les limites de cette méthode d'estimation sont bien connues (Gale *et al.*, 1994). La principale provient du fait que la méthode peut avoir tendance à réserver une masse de probabilité trop importante aux événements non observés et à diminuer ainsi, de façon brutale, les probabilités des événements observés par rapport à leur estimation par maximum de vraisemblance. Malgré cela, nous avons conservé cette méthode d'estimation car, dans notre cas, le nombre d'événements sur lesquels répartir la masse de probabilité est modeste (par comparaison aux événements pris en compte dans l'estimation de n-grams en reconnaissance de la parole, par exemple). Les probabilités attribuées aux événements observés sont par conséquent peu affectées par le lissage des probabilités. Néanmoins, nous avons légèrement modifié la méthode initiale en ajoutant aux comptes une quantité $x \leq 1$. La valeur optimale de x a été déterminée de manière expérimentale sur le corpus de développement.

Probabilités des transitions modificatrices

Les probabilités des transitions modificatrices sont des probabilités conditionnelles d'une catégorie, étant donné la catégorie précédente (ces probabilités sont appelées probabilités bigrammes).

Les probabilités bigrammes sont estimées à partir des comptes contenus dans la table $\mathcal{T}_A$. Pour les mêmes raisons que celles évoquées ci-dessus, l'estimation n'est pas réalisée par maximum de vraisemblance, mais par une variante de la méthode de combinaison linéaire des probabilités bigrammes et unigrammes (la probabilité d'occurrence d'une catégorie indépendamment de la catégorie précédente), proposée par (Jelinek *et al.*, 1980). Cette méthode permet d'estimer la probabilité conditionnelle $P(c_2|c_1)$ comme la combinaison linéaire des deux probabilités $P_{MV}(c_2|c_1)$ et $P_{MV}(c_2)$ (les probabilités estimées par maximum de vraisemblance) :

$$P(c_2|c_1) = \lambda_1 P_{MV}(c_2|c_1) + \lambda_2 P_{MV}(c_2) \text{ avec } \lambda_1 + \lambda_2 = 1 \quad [5]$$

La différence entre la méthode initiale et la nôtre est que, dans la première, les valeurs des coefficients de pondération sont calculés pour chaque historique. Ainsi, l'équation (5) se réécrit :

$$P(c_2|c_1) = \lambda_1(c_1)P(c_2|c_1) + \lambda_2(c_1)P(c_2) \text{ avec } \lambda_1(c_1) + \lambda_2(c_1) = 1 \quad [6]$$

où $\lambda_1(c_1)$ et $\lambda_2(c_1)$ sont les coefficients spécifiques à l'historique c_1. Dans notre cas, une seule valeur a été calculée pour les coefficients λ_1 et λ_2. Ces valeurs ont été déterminées de manière expérimentale sur le corpus de développement.

4.5. *Résultats*

Les expériences ont été effectuées sur la partie du corpus LE MONDE étiquetée syntaxiquement (Abeillé *et al.*, 2003). Le corpus a été divisé en trois parties contiguës : un corpus d'apprentissage (90% du corpus complet), un corpus de test (5%) et un corpus de développement. Le corpus d'apprentissage a servi à l'extraction des grammaires, à la collecte des comptes d'attachement nécessaires à l'estimation des paramètres des modèles probabilistes ainsi qu'à la constitution d'un corpus étiqueté par les catégories définies par les différentes grammaires. C'est sur ce dernier qu'ont été estimés les paramètres d'un étiqueteur grammatical. Le corpus de développement a servi à la mise au point des grammaires, ainsi qu'à l'estimation des coefficients de pondération des probabilités des GDGP-2G. Le corpus de test a été utilisé pour évaluer les performances de l'analyseur et de l'étiqueteur grammatical. Les tailles des trois corpus sont reproduites dans le tableau 1.

Comme nous l'avons évoqué en 4.2, le processus d'extraction de grammaires, proposé par (Chen, 2002), nécessite de distinguer, parmi les dépendants d'un mot, ses

	Apprentissage	Test	Développement
Nombre de phrases	13 716	773	865
Nombre de mots	358 545	19 308	21 401

Tableau 1. *Taille des corpus d'entraînement, de test et de développement*

actants de ses modifieurs. Cette distinction est réalisée dans le Penn Treebank (le corpus dont s'est servi (Chen, 2002)) en utilisant diverses informations, en particulier les étiquettes sémantiques présentes dans le corpus. L'absence d'étiquettes sémantiques, et de rôles fonctionnels, dans le corpus LE MONDE, rend plus difficile la distinction actant/modifieur, en particulier pour les groupes prépositionnels. (Dybro Johansen, 2004) propose trois heuristiques pour contourner le problème. La première, notée H_1, revient à considérer que tous les groupes prépositionnels constituent des modifieurs. H_3 considère, quant à elle, que tous les groupes prépositionnels sont des actants. L'heuristique H_2 constitue une position intermédiaire entre H_1 et H_3. Elle repose sur une liste de 14 prépositions établie manuellement dont on considère qu'elles introduisent des actants[11]. Les 103 autres prépositions (principalement des locutions prépositionnelles) introduisent des modifieurs.

Ces trois heuristiques permettent de générer trois grammaires TIG que nous appellerons respectivement G_1, G_2 et G_3. Ces dernières se distinguent principalement par le nombre de catégories qu'elles définissent. G_3 définit, naturellement, le plus de catégories. En effet, en TAG (et en TIG), les actants sont intégrés dans les arbres élémentaires de leur gouverneur. Par conséquent, plus une grammaire définit d'actants, plus elle produira de schémas d'arbres élémentaires différents.

La distinction actant/modifieur exerce aussi une influence sur l'ambiguïté de la grammaire : en définissant plus d'actants parmi les groupes prépositionnels, une grammaire réduit les cas d'ambiguïté de rattachements prépositionnels. Dans le cas de G_3, par exemple, qui traite tous les groupes prépositionnels comme actants, une catégorie prévoit exactement le nombre de prépositions qu'elle doit régir. A l'inverse, pour G_1, les parties du discours *nom*, *verbe*, *adjectif* et *adverbe* peuvent gouverner un nombre quelconque de prépositions, augmentant ainsi la combinatoire des rattachements prépositionnels.

Une grammaire extraite, G, est évaluée par sa *taille*, sa *couverture* son *ambiguïté lexicale* et son *ambiguïté syntaxique*.

La taille de G n'est autre que le nombre de catégories, ou encore d'automates, qu'elle comprend.

La couverture de G, étant donné un corpus O, est le rapport du nombre d'occurrences de catégories inconnues dans O par la taille de O (le nombre total d'occurrences

11. Ces prépositions sont : *avec, chez, comme, contre, de, en, entre, hors, par, sans, pour, sous, sur, vers.*

de mots dans O). Une catégorie inconnue est une catégorie qui n'a pas été créée lors du processus d'extraction de la grammaire (elle apparaît dans O test mais pas dans le corpus d'apprentissage).

L'ambiguïté lexicale est le nombre moyen de catégories auxquelles est associé un mot dans le lexique.

L'ambiguïté syntaxique de G est la moyenne du nombre d'analyses différentes que G associe aux phrases de O correctement étiquetées (à chaque mot est associé sa catégorie correcte dans le cadre de la phrase).

La taille, couverture, ambiguïté lexicale et ambiguïté syntaxique des trois grammaires sont représentées dans le tableau 2. La couverture et l'ambiguïté syntaxique ont été calculées sur le corpus de test. L'ambiguïté syntaxique a été calculée en réalisant l'analyse syntaxique de chaque phrase (correctement étiquetée) et en comptant le nombre d'analyses obtenues.[12]

	Taille	Couver-ture	Ambiguïté lexicale	Ambiguïté syntaxique	Taille réelle du corpus de test	
					en mots	en phrases
G_1	3 808	$0,993$	$1,859$	11 003	15 161	667
G_2	5 910	$0,988$	$2,089$	750	12 437	586
G_3	7 123	$0,985$	$2,123$	346	11 498	551

Tableau 2. *Taille des trois grammaires G_1, G_2 et G_3 ainsi que leur couverture, leur ambiguïté lexicale, leur ambiguïté syntaxique et la taille réelle du corpus de test sur laquelle a été calculée l'ambiguïté syntaxique*

Les résultats du tableau 2 confirment et quantifient les prédictions théoriques. G_3 définit approximativement deux fois plus de catégories que G_1, et G_2 se situe entre les deux. Malgré les différences importantes de taille des grammaires, leurs ambiguïtés lexicales sont assez proches. Ceci s'explique par le fait que de nombreuses catégories définies par la grammaires sont peu fréquentes (ce phénomène est analysé plus en détails ci-dessous). La comparaison de l'ambiguïté syntaxique pour G_1 et pour G_3 permet de quantifier la part d'ambiguïté provenant des rattachements prépositionnels. En effet, ces deux grammaires ne se distinguent que par leur traitement des rattachements

12. Le corpus sur lequel ont été menées ces expériences ne correspond en fait pas exactement au corpus de test, mais à une partie de ce dernier : la partie constituée des phrases ne contenant pas de catégories inconnues. Rappelons que certaines phrases du corpus de test contiennent des mots dont la catégorie correcte n'a pas été créée lors de l'extraction de la grammaire. Ces phrases ont été éliminées du corpus de test pour ces expériences. La taille du corpus de test amputé de ces phrases est par conséquent inférieure à celle du corpus de test original. Nous l'appellerons taille réelle du corpus, elle est indiquée dans le tableau 2. Cette taille est différente pour les trois grammaires, car chacune possède une couverture différente.

prépositionnels[13], qui ne sont jamais ambigus pour G_3. La différence de l'ambiguïté syntaxique pour G_1 et G_3 est très largement dûe aux rattachements prépositionnels. Parmi les $11\ 003$ analyses construites en moyenne par phrase pour G_1, à peu près $10\ 600$ proviennent de rattachements prépositionnels, soit une proportion de l'ordre de $95\ \%$.

Le tableau 2 permet aussi d'observer que les trois heuristiques exercent une faible influence sur la couverture des grammaires qu'elles définissent : le rapport des tailles des grammaires G_1 et G_3 vaut $0,53$ alors que le rapport de leur couverture n'est que de $1,008$. Un accroissement considérable du nombre de catégories ne diminue donc que très faiblement la couverture des grammaires. Cette différence de comportement s'explique aisément à la lumière de la figure 9, empruntée à (Dybro Johansen, 2004). Les trois graphiques de la figure retracent l'évolution du nombre de catégories définies lors de l'extraction des trois grammaires. Dans chaque graphique, les trois courbes indiquent les catégories observées au moins une fois, au moins deux fois et au moins trois fois, à différentes étapes de l'extraction. On observe qu'à la fin du processus, 63% des catégories de G_3 n'ont été observées qu'une fois (phénomène habituellement désigné par *hapax legomena* (« lu une seule fois », en grec)), 54% dans le cas de G_1. Ces chiffres montrent qu'un nombre important de catégories a une probabilité d'occurrence médiocre. Par conséquent, leur influence sur la couverture de la grammaire est limitée, expliquant la faible influence du nombre de catégories sur la couverture de la grammaire.

Les courbes de la figure 9 montrent aussi que la croissance du nombre de catégories n'est pas stabilisée à l'issue de l'extraction, ce qui semble indiquer que la taille du corpus d'apprentissage n'est pas suffisante pour atteindre la convergence des grammaires. Le résultat pratique de ce fait est que certains arbres élémentaires du corpus de test (corpus sur lequel sont effectuées les expériences d'analyse) n'existent pas dans la grammaire extraite, il s'agit des catégories inconnues évoquées précedemment.

Cette conclusion doit néanmoins être modérée, en observant que le nombre de catégories apparaissant au moins deux fois dans le corpus d'apprentissage est quasiment stabilisé à l'issue de l'extraction. Seules les courbes retraçant l'évolution des *hapax* continuent de croître à l'issue du processus d'extraction. Malgré son influence limitée sur la couverture de la grammaire, l'évolution de la courbe des *hapax* est un phénomène troublant. Il s'explique, en partie, par des erreurs dans le corpus d'apprentissage (une erreur dans un arbre syntaxique du corpus d'apprentissage peut provoquer la création d'un arbre élémentaire aberrant dont la probabilité d'être synthétisé une deuxième fois est quasi-nulle). Ce phénomène n'a pas été quantifié pour l'instant. Les autres *hapax* modélisent des phénomènes rares, susceptibles de se répéter dans un corpus plus important.

13. Ceci n'est pas tout à fait exact. La procédure d'extraction décrite dans (Dybro Johansen, 2004) ne permet pas d'adjoindre un modifieur entre deux actants d'un même gouverneur. Par conséquent, dans la grammaire G_3, les modifieurs situés entre deux groupes prépositionnels - traités comme actants du fait de l'heuristique H_3 - sont aussi traités comme des actants. Ce phénomène est néanmoins marginal et nous le négligerons ici.

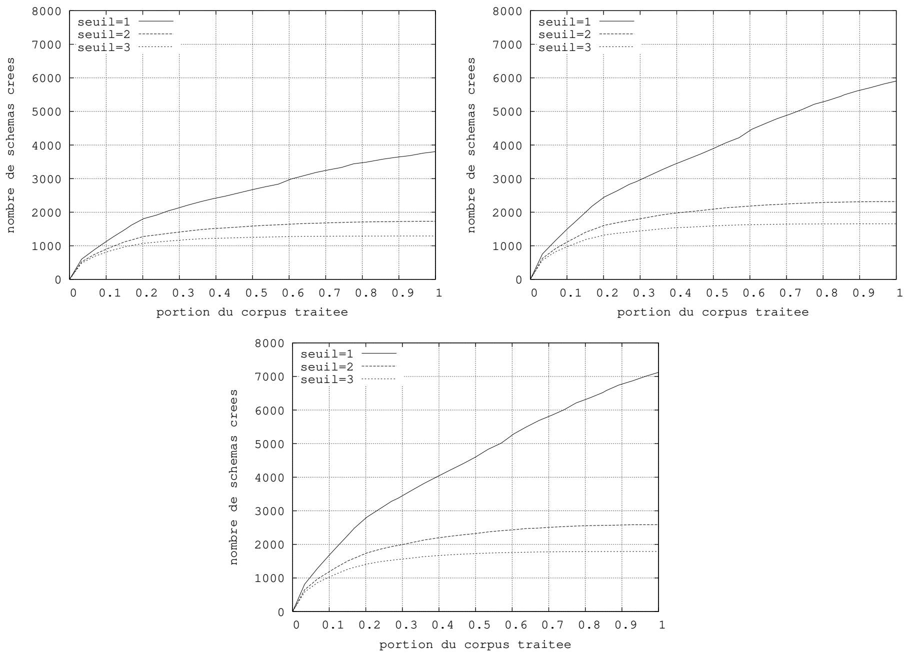


Figure 9. *Croissance de la taille des trois grammaires G_1 (en haut à gauche), G_2 (en haut à droite) et G_3 (en bas) en fonction de la fraction traitée du corpus d'apprentissage. Les trois courbes de chaque graphique représentent les schémas d'arbres observés au moins une fois (seuil = 1), au moins deux fois (seuil = 2) et au moins trois fois (seuil = 3)*

5. Analyse syntaxique

Le processus d'analyse syntaxique pour grammaires GDGP que nous avons élaboré est composé de trois modules : un étiqueteur grammatical, un analyseur et un module de recherche de l'analyse la plus probable parmi les analyses produites. Etant donné une GDGP G, définissant un ensemble de catégories, et une phrase S, l'étiqueteur grammatical produit une ou plusieurs séquences de catégories. L'ensemble des séquences produites est représenté sous la forme d'un automate acyclique, appelé *treillis de catégories*. Chaque chemin d'un tel automate représente une des séquences produites par l'étiqueteur. Cet automate constitue l'entrée de l'analyseur syntaxique, qui produit toutes les analyses possibles de la phrase d'entrée, sous la forme d'une structure compacte, appelée *forêt de dépendance partagée* (FDP). Lorsque l'on fournit à l'analyseur un treillis de catégories ne permettant pas d'aboutir à au moins une analyse complète de la phrase, l'analyseur produit toutes les séquences d'analyses partielles possibles, représentées sous la forme d'une FDP. La FDP produite constitue l'entrée du module de recherche de l'analyse complète (ou de la séquence d'analyses partielles) la plus probable : résultat final de l'analyse.

Comme nous l'avons annoncé dans l'introduction, nous ne décrirons pas les trois algorithmes qui constituent notre chaîne de traitement, dont le lecteur pourra trouver tous les détails dans (Nasr, 2004). Nous décrirons néanmoins, en 5.1, les interactions entre l'étiqueteur et l'analyseur, qui constituent un aspect important de notre système. Les mesures d'évaluation des performances de l'étiqueteur et de l'analyseur seront décrites en 5.2 avant de donner, en 5.3, les résultats de ces modules sur le corpus LE MONDE.

5.1. *Nécessité d'un élagage*

L'analyse syntaxique de phrases extraites de corpus journalistiques à l'aide de grammaires de tailles importantes, telles que les grammaires décrites en 4, peut mener à la construction de centaines de millions d'analyses différentes. Dans le pire des cas, le nombre d'analyses est, en effet, une fonction exponentielle de la longueur de la phrase analysée, comme l'ont montré (Church *et al.*, 1982). Les algorithmes d'analyse syntaxiques standard, en particulier les algorithmes (Younger, 1967) et (Earley, 1968), permettent de construire ces analyses avec une complexité cubique en temps et quadratique en espace, en recourant à des techniques de programmation dynamique. Malgré cela, dans les hypothèses décrites ci-dessus (taille des grammaires et longueur des phrases), la construction de l'intégralité des analyses de certaines phrases ne peut être réalisée dans des conditions raisonnables de temps et d'espace. C'est la raison pour laquelle les analyseurs syntaxiques probabilistes utilisent généralement le modèle probabiliste pour effectuer un élagage en cours d'analyse. L'idée est simple : elle consiste à calculer les probabilités des structures partielles construites aux différentes étapes de l'analyse et à n'en conserver que les plus probables pour poursuivre l'analyse. Une telle technique d'élagage est connue sous le terme anglais de *beam search* et a été introduite à l'origine dans les systèmes de reconnaissance de parole par (Lowerre, 1968). L'avantage de cet élagage est d'augmenter les performances en temps et en espace de l'analyseur. Le risque est d'éliminer, en cours d'analyse, l'analyse partielle qui donnera naissance à la bonne solution.

La présence d'un étiqueteur grammatical, dans notre système, permet de réaliser une autre forme d'élagage : celle qui consiste, comme nous l'avons mentionné dans l'introduction, à ne sélectionner qu'une partie de la grammaire pour mener à bien l'analyse syntaxique. Ce filtrage de la grammaire a une influence importante sur les performances de l'analyseur. En effet, la complexité en temps de ce dernier est en $O(n^3 \times |G|^2 \times E)$ et sa complexité en espace est $O(n^2 \times |G| \times E)$ où n est la longueur de la phrase analysée, $|G|$ est le nombre d'automates de la grammaire et E le nombre maximal d'états par automate de la grammaire. L'étape d'étiquetage permet de diminuer la complexité en temps et en espace mémoire en diminuant le paramètre $|G|$ (qui est alors égal à la taille de la partie de la grammaire sélectionnée par l'étiqueteur).

Dans un système d'analyse syntaxique composé d'un étiqueteur et d'un analyseur, la résolution de l'ambiguïté lexicale et la résolution de l'ambiguïté syntaxique sont effectuées par deux processus différents : l'étiqueteur et l'analyseur. Il est possible de répartir de différentes manières l'ambiguïté totale entre ambiguïté syntaxique et ambiguïté lexicale en faisant varier le nombre de catégories définies par la grammaire. Une grammaire définissant beaucoup de catégories augmentera l'ambiguïté lexicale exigeant ainsi plus de travail de la part de l'étiqueteur tandis que la la tâche de l'analyseur sera allégée. À l'inverse, une grammaire définissant peu de catégories diminuera l'ambiguïté lexicale et augmentera l'ambiguïté syntaxique. Il existe plusieurs moyens de faire varier le nombre de catégories définies par la grammaire. Nous avons vu en 4.5 une première manière de procéder, qui consiste à faire varier les paramètres du processus d'extraction. Les résultats d'analyse et d'étiquetage à l'aide des trois grammaires ainsi produites sont décrits en 5.3. Une autre manière de faire varier le nombre de catégories, plus particulièrement de le réduire, consiste à regrouper certaines catégories au sein de catégories plus générales.

5.2. *Mesures d'évaluation des résultats*

Les expériences décrites dans le reste de cette partie mettent en jeu deux processus : l'étiquetage grammatical et l'analyse syntaxique. Des mesures différentes sont utilisées pour évaluer chacun d'eux.

5.2.1. *Etiquetage*

Le résultat de la tâche d'étiquetage d'un corpus O est évaluée, de manière classique, par la *précision par mot*. Il s'agit du rapport du nombre d'occurrences de O correctement étiquetées par le nombre total d'occurrences de O.

Lorsque l'étiqueteur produit plusieurs solutions, on appelle *précision maximale* de cet ensemble, la précision de la solution la plus proche de la référence.

5.2.2. *Analyse*

Le résultat de l'analyse syntaxique est évalué par deux mesures : la *précision par arbre* (ou A-précision) et la *précision par dépendance* (ou D-précision). Etant donné un corpus O et une grammaire G, la précision par arbre est le nombre de phrases de O pour lesquelles l'arbre de meilleure probabilité est l'arbre correct, divisé par le nombre de phrases de O.

La A-précision est une mesure assez grossière, car elle ne permet pas de savoir si un résultat est partiellement correct. Des mesures plus précises des résultats d'une analyse syntaxique ont été définies pour les arbres syntagmatiques. C'est le cas en particulier des mesures des campagnes d'évaluation PARSEVAL (Black *et al.*, 1991). Ces dernières distinguent *précision*, *rappel* et *croisement*. Etant donné un arbre syntagmatique d'une phrase produit par l'analyseur, que l'on appellera arbre *hypothèse*, et l'arbre correct de cette phrase, appelé arbre *référence*, les trois mesures vont comparer les syntagmes de l'hypothèse et de la référence. Un syntagme est défini par son étiquette et par son extension : les mots qu'il regroupe. La précision d'une hypothèse est définie comme le nombre de syntagmes de l'hypothèse présentes dans la référence. Le rappel est le nombre de syntagmes de la référence présents dans l'hypothèse. Le croisement est le nombre de syntagmes de l'hypothèse qui croisent des syntagmes de la référence ($[_H\ [_R\ ]_H\ ]_R$).

La comparaison d'arbres de dépendances est plus simple que la comparaison d'arbres syntagmatiques, du fait que tous les arbres de dépendances d'une même phrase comportent le même nombre de dépendances ($n - 1$ dépendances pour une phrase composée de n mots). Etant donné un arbre de dépendances hypothèse et un arbre de dépendances référence, on définit la D-précision de l'hypothèse comme le rapport du nombre de dépendances de l'hypothèse présentes dans la référence sur le nombre total de dépendances de l'hypothèse (ou de la référence, les deux étant égaux).

5.3. ***Résultats***

Les expériences décrites ici ont été réalisées sur le corpus LE MONDE à l'aide des GDGP-2G G_1, G_2 et G_3 décrites en 4.5 et du corpus étiqueté produit à l'issue de l'opération de construction des grammaires. Ce corpus a servi à estimer les paramètres de l'étiqueteur grammatical. Nous évaluons en sections 5.3.1 et 5.3.2 l'analyseur et l'étiqueteur pris séparément. Les expériences sur l'analyseur ont consisté à prendre en entrée une phrase correctement étiquetée, à en construire toutes les analyses et à rechercher l'analyse la plus probable. La seule ambiguïté qui est traitée dans ces expériences est donc l'ambiguïté syntaxique. Les expériences sur l'étiqueteur ont consisté à étiqueter une phrase et à étudier l'évolution de la qualité de l'étiquetage en fonction du nombre de solutions produites par l'étiqueteur. C'est donc la seule ambiguïté lexicale qui est traitée. Finalement, en 5.3.3, les deux modules sont évalués conjointement : les sorties de l'étiqueteur sont fournies en entrée à l'analyseur et le résultat fourni par ce dernier est évalué.

5.3.1. *Performances de l'analyseur étant donné un étiquetage correct*

La A-précision et la D-précision calculées sur le corpus de test sont rapportées pour les trois grammaires dans le tableau 3. Ces chiffres ont été calculés sur l'arbre de probabilité maximale de chaque phrase. Les expériences, conformément à l'intuition, montrent que les performances de la grammaire G_3 sont supérieures à celles de G_2, elles-mêmes supérieures à celle de G_1. Ce résultat peut s'expliquer par le fait que la tâche d'analyse est plus facile pour G_3 qu'elle ne l'est pour G_2 et pour G_1. En effet, comme nous l'avons observé en 4.5, le nombre d'analyses différentes par phrase varie en fonction des grammaires. La grammaire G_3 produit en moyenne moins d'analyses que G_2 et que G_1. La tâche consistant à choisir l'analyse correcte, ou une analyse qui s'en rapproche, est par conséquent plus facile pour G_3 qu'elle ne l'est pour G_2 et pour G_1.

Grammaire	A-précision	D-précision
G_1	$0,606$	$0,964$
G_2	$0,682$	$0,971$
G_3	$0,722$	$0,976$

Tableau 3. *A-précision et D-précision des trois grammaires sur le corpus de test, dont la taille réelle dépend de la grammaire*

Les résultats du tableau 3 permettent de comparer entre elles les trois grammaires. Mais ils sont difficiles à évaluer dans l'absolu, sans avoir de point de repère permettant d'estimer la difficulté de la tâche consistant à réaliser l'analyse syntaxique de phrases correctement étiquetées. Nous avons créé un tel point de repère en construisant, pour chaque grammaire G_i, un modèle aléatoire B_i. Comme son nom l'indique, ce dernier a été construit en attribuant aux différents paramètres du modèle des valeurs aléatoires. Les D-précisions obtenues par les trois grammaires B_1, B_2 et B_3 sur les corpus de test sont respectivement égales à $0,934$, $0,951$ et $0,957$.

Ces résultats appellent plusieurs remarques. La première est leur niveau élevé. Il montre que, dans un système où l'analyse syntaxique est réalisée à la suite d'un étiquetage grammatical, l'essentiel des choix a été réalisé lors de l'étiquetage, du moins pour les trois grammaires G_1, G_2 et G_3. On observe, en effet, qu'en choisissant la catégorie des mots d'une phrase et, par conséquent, la grammaire utilisée lors de l'analyse syntaxique, l'ambiguïté syntaxique est suffisamment limitée pour qu'un modèle aléatoire effectue les bons rattachements, dans 95% des cas. Cette conclusion permet de quantifier la prédiction de S. Bangalore et A. Joshi : « supertagging is almost parsing ». La valeur de *almost* a donc été estimée comme étant égale à 95%. D'autre part, on peut observer que les gains apportés par les modèles probabilistes sont faibles : les résultats de G_1 sont supérieurs à ceux de B_1 de $3,2\%$. La différence est de 2% pour le couple (G_2, B_2) et de $1,9\%$ pour le couple (G_3, B_3)[14]. Nous verrons en 5.3.3 que l'apport des modèles probabilistes est nettement supérieur dans des conditions d'analyse plus réalistes.

5.3.2. *Performances de l'étiqueteur*

L'étiqueteur utilisé dans ces expériences est un étiqueteur trigramme standard, fondé sur le modèle des chaînes de Markov cachées. Nous avons représenté, dans la partie gauche de la figure 10, la précision maximale de l'étiquetage, pour les trois grammaires G_1, G_2 et G_3, en fonction du nombre de solutions demandées à l'étique-

14. Etant donné la taille réduite des échantillons sur lesquels ces scores ont été calculés, nous ne pouvons dire si les différences observées sont significatives. Il aurait été nécessaire, pour cela, de recourir à un test de significativité.

teur. Rappelons que la précision maximale est la précision de la meilleure solution fournie par l'étiqueteur lorsque celui-ci en produit plusieurs.

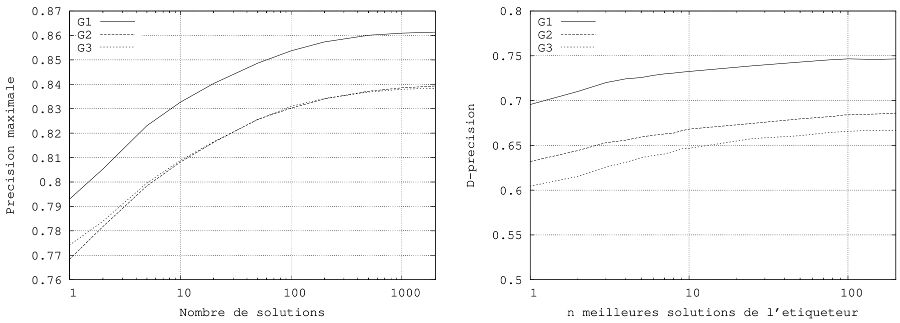


Figure 10. *À gauche : Précision de l'étiquetage en fonction du nombre d'hypothèses produites. À droite : A-précision et D-précision de la solution la plus probable de l'analyseur en fonction du nombre de solutions de l'étiqueteur grammatical*

On observe sans surprise que la précision dépend du nombre de catégories de la grammaire, mais elle dépend aussi de la nature de ces catégories. En effet, les résultats des deux grammaires G_2 et G_3 sont très proches, bien que G_3 possède approximativement 20% de catégories de plus que G_2. On observe aussi que la précision maximale n'augmente quasiment plus à partir de 200 solutions. Les résultats obtenus sont nettement inférieurs à ceux obtenus par S. Bangalore sur le corpus du Penn Treebank ($92,2\%$ pour la meilleure solution de l'étiqueteur). La différence majeure entre les expériences de S. Bangalore et les expériences décrites ici concerne le nombre de catégories : 300 dans le premier cas et, respectivement, 3 808, 5 910 et 7 123 dans le second. Cette différence explique probablement l'écart entre les résultats obtenus par S. Bangalore et les résultats présentés ici.

5.3.3. *Performances de l'analyseur couplé à un étiqueteur*

Les expériences d'analyse de la section 5.3.1 avaient été effectuées dans des conditions artificielles : celles d'un étiquetage grammatical parfait. Il s'agit, en quelque sorte des meilleurs résultats que l'on peut obtenir à l'aide des trois grammaires G_1, G_2 et G_3. Les expériences décrites dans cette section sont plus proches des conditions réelles. En effet, l'entrée du système est constituée de l'intégralité des phrases du corpus de test. Ces phrases sont traitées, dans un premier temps, par un étiqueteur grammatical produisant un certain nombre de solutions (fixé par l'utilisateur), représentées sous la forme d'un treillis de catégories. Le treillis est ensuite fourni en entrée à l'analyseur syntaxique.

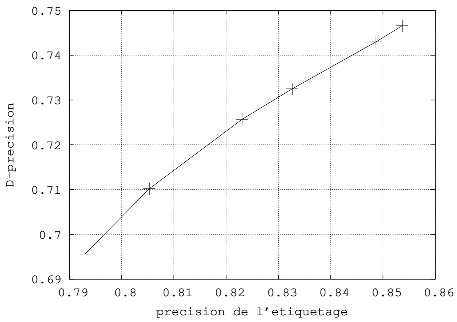
La D-précision de la solution la plus probable de l'analyseur, en fonction du nombre de solutions produites par l'étiqueteur, pour les trois grammaires, est représentée dans la partie droite de la figure 10. Les meilleurs résultats sont obtenus pour la grammaire G_1, alors que G_3 obtenait les meilleurs résultats pour une entrée correctement étiquetée. Les résultats de l'analyse couplée à l'étiquetage suivent donc la tendance observée sur l'étiquetage. Le meilleur résultat obtenu pour G_1 est de $0,746$ (D-précision) en prenant en entrée de l'analyseur le treillis des 200 meilleures solutions de l'étiqueteur, dont la précision maximale est égale à $0,853$.

Les résultats obtenus sont très nettement inférieurs à ceux obtenus sur des phrases correctement étiquetées pour la même grammaire. Ces derniers étaient de $0,964$ (D-précision). Cette différence permet de quantifier l'influence des erreurs d'étiquetage sur les performances de l'analyse : une diminution de 15% de précision de l'étiqueteur provoque une diminution de $22,4\%$ de la D-précision de l'analyse[15].

Afin de mieux apprécier l'influence des erreurs d'étiquetage sur les erreurs d'analyse, nous avons tracé, dans la partie gauche de la figure 11, la D-précision du résultat de l'analyse par rapport à la précision de l'étiquetage pour la grammaire G_1. Les différentes précisions d'étiquetage ont été obtenues en faisant varier le nombre de solutions produites par l'étiqueteur. Cette figure montre que la relation entre les deux variables est assez proche d'une relation linéaire, sur l'intervalle que nous avons à notre disposition (une précision de l'étiqueteur variant de $0,79$ à $0,85$). Cet intervalle et le nombre de points dont on dispose est malheureusement trop limité pour effectuer une extrapolation et déterminer la précision d'étiquetage nécessaire pour obtenir une D-précision donnée.

Afin d'évaluer la difficulté de la tâche d'analyse prenant en entrée un treillis de catégories, nous avons eu recours aux modèles aléatoires introduits dans la section précédente. Les D-précisions obtenues par la grammaire aléatoire B_1 (équivalente à la grammaire G_1 mais dont les paramètres ont été déterminés aléatoirement) sont représentées dans la partie droite de la figure 11. La D-précision de G_1 a été reprise dans la figure pour faciliter la comparaison des deux courbes. On observe que les performances du modèle aléatoire diminuent lorsque l'on augmente le nombre de solutions produites par l'étiqueteur. L'apport du modèle probabiliste est ici nettement plus important qu'il ne l'était pour une entrée correcte. En effet, le gain varie entre $14,8\%$ et $30,4\%$ alors qu'il était de $3,2\%$ pour une entrée correcte. En définitive, le modèle probabiliste semble avoir un faible pouvoir discriminant lorsque les phrases ont été correctement étiquetées : la solution qu'il propose n'est pas très éloignée d'une solution choisie au hasard. En revanche, lorsque l'entrée de l'analyseur est constituée de plusieurs séquences possibles d'étiquetage (un treillis de catégories) la solution pro-

15. Cette comparaison est un peu biaisée du fait que, dans le premier cas, seule la bonne séquence de catégories était fournie à l'analyseur, alors qu'ici, un treillis de 200 séquences de catégories lui est fourni. On ne sait, à l'issue de l'analyse, si la solution produite par l'analyseur correspond à la meilleure séquence de catégories du treillis. Pour le déterminer, nous avons extrait de la solution de l'analyseur la séquence de catégories à laquelle elle correspond : les catégories qui ont permis de constituer la solution proposée par l'analyseur. Nous avons observé que cette séquence correspondait, dans une très large mesure, à la meilleure séquence. L'analyseur est donc un bon étiqueteur !



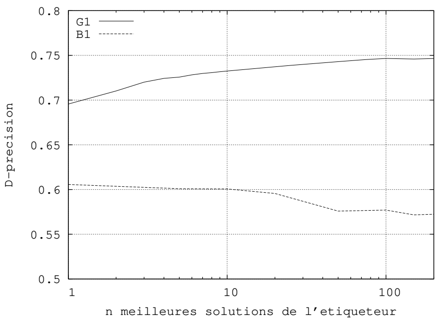


Figure 11. *À gauche : D-précision de l'analyseur en fonction de la précision de l'étiqueteur, pour la grammaire* G_1*. À droite : A-précision et D-précision de la solution la plus probable de l'analyseur avec le modèle bigramme et avec le modèle aléatoire de la grammaire* G_1*. L'entrée de l'analyseur est constituée des* n *solutions les plus probables de l'étiqueteur*

posée par le modèle probabiliste est nettement meilleure qu'une solution choisie au hasard.

6. Conclusion et perspectives

Nous avons décrit dans cet article un formalisme pour grammaires de dépendances, appelé GDGP et avons rendu compte d'un certain nombre d'expériences effectuées sur le corpus LE MONDE. En guise de conclusion, nous allons revenir sur un aspect théorique des GDGP, l'adéquation entre le modèle algébrique et probabiliste, ainsi que sur un aspect expérimental, la comparaison des résultats obtenus ici et ceux obtenus sur le corpus du Penn Treebank.

D'un point de vue théorique, une des caractéristiques des GDGP est l'adéquation qu'elles pemettent entre un modèle algébrique et un modèle probabiliste. Nous avons vu en effet comment les hypothèses d'indépendances probabilistes sont matérialisées par la structure des automates de la grammaire. La principale raison qui nous a poussé à rechercher une telle adéquation est la volonté d'avoir recours, quelle que soit le type de GDGP utilisé, à un seul algorithme d'analyse et à un seul algorithme de recherche de la meilleure solution dans la forêt produite à l'issue de l'analyse. Cependant, cette caractéristique possède des inconvénients. En particulier lorsque le modèle probabiliste introduit des hypothèses ne pouvant être matérialisées sous la forme d'un automate. On pense en particulier à la possibilité de conditionner la probabilité d'un attachement à d'autres ancêtres d'un mot que son gouverneur direct. Par exemple le gouverneur du gouverneur. Les travaux décrits dans (Klein *et al.*, 2003) ont en effet montré que ce genre de probabilité pouvait accroître le pouvoir discriminant d'une grammaire probabiliste. Une telle hypothèse, appelée par Klein et Manning *markovisation verticale* ne peut être simplement implémentée dans une GDGP.

D'un point de vue expérimental, les expériences effectuées dans la section précédente ont permis d'évaluer le système décrit dans cet article. Les meilleurs résultats obtenus sont de $0,746$ (D-précision) sur le corpus LE MONDE. Il est difficile d'évaluer ces résultats dans l'absolu, car nous ne connaissons pas d'expériences analogues sur ce corpus.

Cependant, ainsi que nous l'indiquions dans l'introduction, des expériences analogues ont été effectuées sur le Penn Treebank et ont abouti à une D-précision de l'ordre de $0,9$. Trois différences importantes entre les conditions expérimentales du travail décrit ici et de celui effectué sur le Penn Treebank permettent d'apporter des éléments d'explication à cette différence de performances.

La première est la taille du corpus. Le Penn Treebank, en effet, est constitué d'un million de mots, soit plus du double du corpus que nous avons utilisé. La seconde est la richesse d'étiquetage. Comme nous l'avons évoqué dans la section 4.5, le Penn Treebank définit un certain nombre d'étiquettes sémantiques. Ces dernières permettent de mieux effectuer la distinction actant/modifieur lors de l'extraction de grammaires, aboutissant à des grammaires plus compactes. La troisième différence est la technique utilisée pour réaliser l'étiquetage syntaxique. Les expériences effectuées sur le Penn Treebank ont été réalisées à l'aide d'un étiqueteur fondé sur le principe du maximum d'entropie (Bangalore *et al.*, 2005).

La poursuite du travail d'annotation syntaxique du corpus LE MONDE, tant du point de vue qualitatif que quantitatif va permettre de diminuer certaines différences expérimentales décrites ci-dessus. A terme, en effet, le corpus LE MONDE aura une taille proche de celle du Penn Treebank. De plus, les dernières versions de ce corpus ont été enrichies d'un étiquetage fonctionnel qui permettra une meilleure distinction actant/modifieur, ce qui devrait aboutir à des GDGP de meilleure qualité. Finalement, on peut espérer que les techniques d'étiquetage décrites dans (Bangalore *et al.*, 2005), lorsqu'elles seront mises en œuvre sur le corpus LE MONDE, aboutiront à de meilleurs résultats que l'étiqueteur à base de chaines de Markov cachées utilisé ici.

7. Bibliographie

Abeillé A., Clément L., Toussenel F., « Building a treebank for French », *in* A. Abeillé (ed.), *Treebanks*, Kluwer, Dordrecht, 2003.

Abney S., « A Grammar of Projections », 1996, manuscrit non publié, Universität Tübingen.

Allauzen C., Mohri M., Roark B., « Generalized Algorithms for Constructing Statistical Language Models », *41st Meeting of the ACL*, Sapporo, Japon, p. 40-47, 2003.

Alshawi H., « Head Automata and Bilingual Tiling : Translation with Minimal Representation », *34th Meeting of the ACL'96*, p. 167-176, 1996.

Bangalore S., Complexity of Lexical Descriptions and its Relevance to Partial Parsing, PhD thesis, University of Pennsylvania, Philadelphia, USA, March, 1997.

Bangalore S., Haffner P., Emami G., Factoring Global Inference by enriching local representations, Technical report, AT&T Labs - Research, 2005.

Bangalore S., Joshi A. K., « Supertagging : An Approach to Almost Parsing », *Computational Linguistics*, vol. 25, n° 2, p. 237-265, 1999.

Bikel D., « Intricacies of Collins' Parsing Model », *Computational Linguistics*, vol. 30, n° 4, p. 479-511, 2004.

Black E., Abney S. P., Flickinger D., Gdaniec C., Grishman R., Harrison P., Hindle D., Ingria R., Jelinek F., Klavans J., Liberman M., Marcus M. P., Roukos S., Santorini B., Strzalkowski T., « A Procedure for Quantitatively Comparing the Syntactic Coverage of English Grammars », *Proceedings DARPA Speech and Natural Language Workshop*, Morgan Kaufmann, Pacific Grove, CA, p. 306-311, February, 1991.

Black E., Jelinek F., Lafferty J. D., Magerman D. M., Mercer R. L., Roukos S., « Towards History-based Grammars : Using Richer Models for Probabilistic Parsing », *Proceedings DARPA Speech and Natural Language Workshop*, Morgan Kaufmann, Harriman, New York, p. 134-139, 1992.

Booth T. L., « Probabilistic representation of formal languages », *IEEE Conference Record of the 1969 Tenth Annual Symposium on Switching and Automata Theory*, p. 74-81, 1969.

Charniak E., « Statistical parsing with a context-free grammar and word statistics », *AAAI-97*, AAAI Press, Menlo Park, 1997.

Chen J., Towards Efficient Statistical Parsing using Lexicalized Grammatical Information, PhD thesis, University of Delaware, 2002.

Church K. W., Patil R., « Coping with Syntactic Ambiguity », *American Journal of Computational Linguistics*, vol. 8, n° 3-4, p. 139-149, 1982.

Clark S., « Supertagging for Combinatory Categorial Grammar », *Proceedings of the 6th International Workshop on Tree Adjoining Grammars and Related Frameworks (TAG+6)*, Venise, Italie, p. 19-24, 2002.

Collins M., Head-driven Statistical Models for Natural Language Parsing, PhD thesis, University of Pennsylvania, Philadelphia, 1999.

Collins M., « Head-Driven Statistical Models for Natural Language Parsing », *Computational Linguistics*, vol. 29, n° 4, p. 589-637, 2003.

Dybro Johansen A.*, « Extraction automatique de grammaires à partir d'un corpus français »*, Master's thesis, Université Paris 7, 2004.

Earley J., An Efficient Context-Free Parsing Algorithm, PhD thesis, Carnegie Mellon University, Pittsburgh, PA, 1968.

Eisner J., « Three New Probabilistic Models for Dependency Parsing : An Exploration », *Proceedings of the 17th International Conference on Computational Linguistics (COLING'96)*, p. 340-345, 1996.

Eisner J., « Bilexical Grammars and Their Cubic-Time Parsing Algorithms », *in* H. Bunt, A. Nijholt (eds), *Advances in Probabilistic and Other Parsing Technologies*, vol. 16 of *Text, Speech and Language Technology*, Kluwer Academic Publishers, Dordrecht/Boston/London, p. 29-61, 2000.

Gaifman H., « Dependency Systems and Phrase-Structure Systems », *Information and Control*, vol. 8, p. 304-337, 1965.

Gale W. A., Church K. W., « What is wrong with adding one ? », *in* N. Oostdijk, P. de Haan (eds), *Corpus-based Research into Language*, Rodopi, Amsterdam, p. 189-198, 1994.

Gildea D., « Corpus variation and parser performances », *Proceedings of the 2001 Conference on Empirical Methods in Natural Language Processing*, Pittsburgh, USA, 2001.

Hays D. G., « Dependency Theory : A Formalism and some Observations », *Language*, vol. 40, p. 511-525, 1964.

Jelinek F., Mercer R. L., « Interpolated estimation of Markov source parameters from sparse data », *in* E. S. Gelsema, L. N. Kanal (eds), *Proceedings, Workshop on Pattern Recognition in Practice*, North Holland, Amsterdam, p. 381-397, 1980.

Joshi A., Levy L., Takahashi M., « Tree Adjunct Grammars », *J. Comput. Syst. Sci.*, vol. 10, p. 136-163, 1975.

Jurafsky D., Martin J., *Speech and Language Processing*, Prentice Hall, Upper Saddle River, New Jersey, Etats Unis, 2000.

Klein D., Manning C. D., « Accurate Unlexicalized Parsing », *41st Meeting of the ACL*, 2003.

Lowerre B. T., The Harpy Speech Recognition System, PhD thesis, Carnegie Mellon University, Pittsburgh, PA, 1968.

Magerman D. M., « Statistical Decision-Tree Models for Parsing », *ACL-95*, Cambridge, MA, p. 276-283, 1995.

Mel'čuk I. A., *Dependency Syntax : Theory and Practice*, State University of New York Press, New York, 1988.

Nasr A., « Analyse syntaxique probabiliste pour grammaires de dépendances extraites automatiquement », , Habilitation à diriger des recherches, Université Paris 7, décembre, 2004.

Nasr A., Estève Y., Béchet F., Spriet T., de Mori. R., « A Language Model Combining N-grams and Stochastic Finite State Automata », *Eurospeech*, vol. 5, Budapest, Hungary, p. 2175-2178, 1999.

Nasr A., Rambow O., *Non-lexical chart parsing for TAG*, MIT Press, chapter Complexity of Lexical Descriptions and its Relevance to Natural Language Processing : A Supertagging Approach, 2006.

Resnik P., « Probabilistic Tree-Adjoining Grammar as a Framework for Statistical Natural Language Processing », *COLING-92*, Nantes, France, p. 418-424, 1992.

Schabes Y., Computational and Mathematical Properties of Lexicalized Grammars, PhD thesis, Université de Pennsylvanie, Philadelphie, États-Unis, 1990.

Schabes Y., Waters R. C., « Tree Insertion Grammars : A Cubic-Time, Parsable Formalism that Lexicalizes Context-Free Grammar without Changing the Trees Produced », *Computational Linguistics*, 1995.

Tesnière L., *Eléments de syntaxe structurale*, Klincksieck, Paris, 1959.

Woods W. A., « Transition Network Grammars for Natural Language Analysis », *Transactions of the ACM*, vol. 13, n° 10, p. 591-606, October, 1970.

Younger D., « Recognition and parsing of context-free grammar in time n^3 », *Information and Control*, vol. 10, p. 189-208, 1967.