Induction d'une grammaire de propriétés à granularité variable à partir du treebank arabe ATB

Raja Bensalem Bahloul[1, 2], Marwa Elkarwi[1, 2]
(1) Laboratoire Mir@cl, FSEGS, Université de Sfax, Sfax, Tunisie
(2) Laboratoire Parole et Langage (LPL), CNRS, Université d'Aix-Marseille, Aix-en-Provence, France
raja_ben_salem@yahoo.com, marwaelkarwi89@gmail.com

Résumé. Dans cet article, nous présentons une démarche pour l'induction d'une grammaire de propriétés (GP) arabe en utilisant le treebank ATB. Cette démarche se base sur deux principales étapes : (1) l'induction d'une grammaire hors contexte et (2) l'induction d'une GP par la génération automatique des relations qui peuvent exister entre les unités grammaticales décrites dans la CFG. Le produit obtenu constitue une ressource ouvrant de nouvelles perspectives pour la description et le traitement de la langue arabe.

Abstract. This paper presents an approach for building an Arabic property grammar using the treebank ATB. This approach consists in two main steps: (1) inducing a context-free grammar from a treebank and (2) inducing a property grammar. So, we acquire first a context-free grammar (CFG) from the source treebank and then, we induce the property grammar by generating automatically existing relations between grammatical units described in the CFG. The result is a new resource for Arabic, opening the way to new tools and descriptions.

Mots-clés : Treebanks, langue arabe, grammaire hors-contexte, grammaires de propriétés
Keywords: Treebanks, Arabic language, context-free grammar, property grammars

1 Introduction

Le formalisme de Grammaires de Propriétés (GP) est l'une des approches linguistiques qui mettent la notion de contraintes au cœur de l'analyse (Blache, 2001 ; Blache, 2005). Ce formalisme se distingue des autres approches basées sur les contraintes par sa représentation simple, directe, locale et décentralisée des informations linguistiques. En effet, à la différence des théories génératives, cette approche représente, d'une manière indépendante, tout type d'information, quelle que soit sa position, mais également des informations incomplètes, partielles et non canoniques, ce qui favorise sa flexibilité et sa robustesse. De plus, cette approche ne requiert pas la construction d'une structure locale des informations syntaxiques avant d'utiliser les contraintes qu'elle a décrites, comme le font les autres approches basées sur les contraintes (en HPSG (cf. (Pollard, 1994)) un arbre local et en CDG (cf. (Maruyama, 1990)) une relation de dépendance). Mais plutôt, elle spécifie les informations syntaxiques directement sur des catégories. Ainsi, une GP est formée par un ensemble de propriétés indépendantes l'une de l'autre. Ces propriétés expriment différentes relations (syntaxiques, sémantiques, etc.) entre les catégories qui forment la structure syntaxique. Elles peuvent être très spécifiques (concernant un ensemble limité de catégories) ou au contraire très générales.

Les qualités qui caractérisent le formalisme de GP nous ont incités à l'utiliser pour construire une nouvelle ressource robuste et riche pour la langue arabe. Le traitement de cette langue présente plusieurs défis. Ces défis ne sont pas seulement liés à certaines spécificités de l'arabe à étudier (comme l'absence des voyelles, la nature agglutinative des mots), mais aussi à des phénomènes linguistiques particuliers à traiter (comme les relatives, les anaphores et les coordinations). Une construction manuelle de cette nouvelle ressource en se basant sur un corpus regroupant toutes les règles de la grammaire arabe, est certainement difficile et couteuse. Ceci requiert en effet beaucoup de temps ainsi que la collaboration de plusieurs linguistes. Une autre manière de procéder peut être proposée : construire la GP à partir d'un corpus annoté. Les treebanks qui sont des corpus annotés manuellement formant une structure morphosyntaxique à plusieurs niveaux d'analyse (niveau mot, niveau syntagme et niveau phrase) peuvent être exploités dans ce cadre. Les treebanks arabes sont déjà rares, le premier lancé étant le treebank PATB (notamment appelé ATB) (Maamouri et Bies, 2004). Nous avons choisi de l'utiliser vu ses qualités qui s'avèrent convenables à la construction de notre GP. La première qualité caractérisant ce treebank est sa représentation à base de syntagmes, conforme à la structure syntaxique

hiérarchisée de la GP à construire. Ce choix est motivé également par la richesse, la fiabilité et la compatibilité des annotations de Part-of-Speech (POS) et de relations syntaxiques et sémantiques de l'ATB à des consensus. Ces annotations sont en fait élaborées et validées par des linguistes. De même, la grammaire de ce treebank est adaptée à l'arabe standard moderne. Il ne faut pas oublier aussi la pertinence, la variété et la grande taille qui caractérisent ses documents sources. Ces documents méritent d'être qualifiés « pertinents » grâce à leur conversion par plusieurs autres treebanks à leur représentation. Le fait de disposer d'une ressource de ce type permet de générer automatiquement et de façon très contrôlée de nouvelles ressources dans d'autres formalismes. Des ressources à large couverture sont ainsi obtenues, héritant des qualités du treebank d'origine tout en gagnant en temps de construction.

Toutefois, le fait que les catégories représentées dans le treebank se caractérisent par une forte granularité, peut affecter la taille des informations à représenter dans la GP. Il faut alors intégrer des mécanismes de contrôle la réduisant. Une autre difficulté peut être rencontrée au niveau de la génération des propriétés dans notre GP. En effet, il y a des propriétés faciles à déduire, mais il y en a d'autres nécessitant des heuristiques.

Dans cet article, le processus d'induction de notre GP se déroule sur deux phases : La première consiste à induire une grammaire hors-contexte (Context-Free Grammar, CFG) à partir de l'ATB. La seconde phase porte sur la déduction des différentes relations qui existent entre les catégories de chaque unité syntaxique à partir des règles de la CFG obtenue. La taille de la GP obtenue peut être contrôlée en variant les différents niveaux de granularité des catégories grammaticales de l'ATB. En plus, avec les types de propriétés définis dans le présent article, la démarche d'induction de GP que nous avons adoptée est purement automatique et indépendante de toute langue et du formalisme du treebank source. Ceci favorise sa réutilisation. Selon nos connaissances, la GP que nous avons obtenue, induite à partir d'un treebank, représente le premier essai produit pour l'arabe.

Cet article est organisé comme suit : la section 2 est consacrée à une brève présentation de l'état de l'art. Ensuite, l'étude de l'ATB est l'objet de la section 3. La section 4 décrit ensuite la démarche d'induction que nous proposons. La section 5 présente les expérimentations et les résultats obtenus de l'application de notre démarche. La section 6 termine par une conclusion et des perspectives.

2 Etat de l'art

Pour pouvoir aborder la problématique de notre travail, nous avons mené des recherches sur deux volets différents : un aperçu sur les approches d'induction de GP et une observation des différentes améliorations effectuées sur l'ATB.

D'une part, ce qu'il y a en commun dans les approches d'induction de GP est leur entrée qui est la CFG. Leurs formalismes sources ou leurs usages quant à eux diffèrent d'une approche à une autre. Concernant l'entrée de ces approches, elle est sous forme de suites d'étiquettes décrivant les unités syntaxiques observées dans un corpus annoté (étiqueté). Ces suites sont en fait représentées par une CFG. Il est vrai que l'entrée, étant la CFG, est commune à toutes les approches d'induction de GP, mais son induction elle-même à partir d'un treebank peut être faite selon des techniques différentes. En effet, elle peut être une CFG simple comme dans (Marcus et al. 1993 ; Hajic, 1998 ; Abeillé et al., 2003 ;Telljohann et al., 2004), ou bien une CFG intégrant des ajustements spécifiques aux suites d'étiquettes. C'est le cas notamment des CFG probabilistes affectant des probabilités à chacune des suites d'étiquettes obtenues, comme dans (Charniak, 1996 ; Mohri et Roark, 2006 ; Rebein et VanGenabith, 2007 ; Tounsi et VanGenabith, 2010). Il existe également plusieurs exemples montrant la différence entre les approches d'induction des GP au niveau de leurs formalismes sources et de leurs usages : Dans (Blache et al., 2003) par exemple, les auteurs ont préparé leur propre corpus étiqueté à partir d'un corpus français brut en passant successivement par une étape de segmentation et une étape d'étiquetage. Chacune de ces deux étapes ont recours à un dictionnaire constitué plus particulièrement d'un lexique appelé DicoLPL et composé d'environ 450 000 formes[1]. La ressource que représente DicoLPL est constituée sur la base d'un lexique interne au LPL[2] et complétée en s'appuyant sur des ressources existantes et des ressources acquises manuellement ou automatiquement par vérification sur corpus. La GP obtenue a été utilisée ensuite dans le cadre d'analyseurs syntaxiques à granularité variable (VanRullen et al., 2005). La base de données Aix-MARSEC (Auran et al., 2004) a formé aussi un formalisme source pour l'induction des GP. En effet, Aix-MARSEC est formée de deux principaux composants : les enregistrements numérisés du corpus MARSEC[3] et leurs annotations. Ces annotations ont

[1] Une version évoluée du lexique DicoLPL, avec plus de formes, a été présenté dans (VanRullen et al., 2005).

[2] www.lpl.univ-aix.fr/

[3] MARSEC (Machine Readable Spoken English Corpus) contient des enregistrements acoustiques numérisés et c'est une extension du corpus SEC disponible en version treebank et en version étiquetée (www.comp.leeds.ac.uk/ccalas/tagsets/sec. html).

été présentées au début à neuf niveaux différents (tels que le niveau phonèmes, syllabes, mots, etc). A ces niveaux, deux niveaux supplémentaires ont été spécifiés : l'annotation syntaxique ainsi qu'un système de GP relatif. De plus, les treebanks représentent également un autre formalisme source pour l'induction des GP. Ainsi, dans (Blache et Rauzy, 2012) par exemple, les auteurs ont bénéficié de ces qualités en utilisant un sous-ensemble du treebank français FTB[4] pour induire leur GP. Ils ont effectué des modifications sur ce sous-ensemble pour assurer une meilleure homogénéité avec les ressources existantes dans d'autres langues ou pour d'autres domaines. Ces modifications sur les niveaux morphologique et syntaxique et sur les positions des marqueurs de ponctuation. La grammaire obtenue a été exploitée ensuite pour enrichir automatiquement le treebank source par une représentation à base de contraintes (Blache et Rauzy, 2012) tout en appliquant un ensemble de solveurs de contraintes. Les travaux d'induction de GP ne se limitent pas uniquement au treebank français (Blache et Rauzy, 2012), mais aussi au treebank chinois (CTB) (Blache, 2014).

D'autre part, l'ATB a été également enrichi en lui intégrant différentes améliorations et corrections pour pouvoir surmonter les défis liés à certaines spécificités de la langue arabe. En effet, cette langue est caractérisée par sa morphologie complexe. Ce problème a été examiné au niveau de l'ATB par (Kulick et al., 2010) en représentant les mots ayant une forme agglutinative par des unités séparées dans une structure arborescente. Par exemple, le mot arabe « كتبه » (ktbh/ *ses livres*), s'il n'est pas voyellé, l'ATB le représente en deux parties tels que « ktb » (*livres*) est un groupe nominal et « h » (*ses*) est un pronom possessif. Pour réaliser cette tâche, les auteurs ont utilisé l'outil SAMA (Standard Morphological Analyzer) pour générer des solutions d'analyse morphologique pour chaque mot de l'ATB. De même, l'absence de voyelles dans la langue arabe peut générer des ambigüités. Pour surmonter cette difficulté, les auteurs de (Kulick et al., 2010) ont intégré une représentation syntaxique abstraite de la structure arborescente tout en autorisant le passage entre les différents niveaux de représentation syntaxique et tout en fournissant différents niveaux de voyellation pour chaque mot de l'ATB. Concrètement, la procédure d'annotation morphosyntaxique que les auteurs ont suivie est basée sur deux grandes étapes : la première consiste en une annotation syntaxique décomposant le texte de l'ATB en mots (appelés jetons sources). Ces mots sont intégrés dans l'outil SAMA, puis générés sous forme voyellée. La deuxième étape, quant à elle consiste à séparer ces jetons des pronoms liés durant l'annotation syntaxique. Par exemple, l'analyse du mot arabe « كتبه » par l'outil SAMA génère une solution qui le décompose en trois segments. Cette solution inclut une séquence d'informations pour chaque segment portant sur trois champs : la forme voyellée, l'étiquette Part-Of-Speech (POS) et la traduction du segment. La solution SAMA de ce mot est présentée comme suit :

[kutub, NOUN, books]	[i, CASE_DEF_GEN, def.gen]	[hi, POSS_PRON_3MS, its/his]

Les auteurs de (Kulick et al., 2010) ont cherché également des procédures spécifiques pour traiter les mots arabes ayant un caractère particulier, tels que les mots ayant une forme agglutinative ne pouvant pas être explicitement décomposée. Le mot عما (EmA/*de ce que*) par exemple est une préposition suivie d'un pronom relatif. La solution proposée par SAMA est composée de deux segments. Elle inclut un « n » dans « Ean », n'ayant pas été présent dans le mot source « EmA ».

[Ean, PREP, from/about/of]	[mA, REL_PRON, what]

Par ailleurs, l'application d'une analyse statistique apprise sur le treebank arabe (ATB) et examinant des incohérences dans les annotations a généré des scores d'analyse inférieurs aux prévisions. Ces incohérences résident au niveau de certaines constructions syntaxiques ou de la relation entre les étiquettes POS et les annotations syntaxiques. La résolution de ces incohérences va corriger largement les directives d'annotation. Le travail (Maamouri et al., 2008) s'inscrit dans ce cadre. En effet, il présente des corrections et des améliorations des incohérences d'annotation dans le but d'améliorer la qualité d'analyse du corpus de l'ATB. Ce travail utilise les étiquettes POS pour corriger et améliorer les directives d'annotation syntaxique. Ces corrections sont proposées aussi bien au niveau morphologique qu'au niveau syntaxique. Au niveau morphologique, les auteurs de (Maamouri et al., 2008) ont proposé de raffiner les étiquettes POS des noms et des adjectifs pour spécifier les noms quantifieurs (NOUN_QUANT), les nombres (NOUN_NUM), les adjectifs comparatifs (ADJ_COMP), les nombres ordinaux (ADJ_NUM), etc. De même, ils ont distingué des catégories d'étiquettes POS pour les différentes particules, telles que l'étiquette CONJ qui a été décomposée en quatre catégories. Et pour distinguer les pseudo-verbes des verbes, ils ont ajouté l'étiquette PSEUDOVERB pour les sœurs de la particule arabe « إنّ » (inna / *que*). Au niveau syntaxique, les auteurs de (Maamouri et al., 2008) se sont focalisés sur la désignation du nom par une étiquette spécifique s'il est un quantifieur dans le syntagme « idafa » pour spécifier correctement la tête sémantique du syntagme. En effet, pour le syntagme « كل مجموعة » (*chaque collection*), la tête sémantique n'est pas le segment « chaque » mais plutôt le segment « collection » parce que celle-là est un nom quantifieur et non pas un simple nom. Il faut alors le spécifier par l'étiquette NOUN_QUANT. Les auteurs de (Maamouri et al., 2008) ont marqué aussi les gérondifs et les participes, si ceux-ci présentent une lecture verbale, par des étiquettes regroupant toute l'unité syntaxique. Par exemple, la phrase « احتفل الفريق بفوزه بكأس الأبطال» (*L'équipe a célébré son gain de la coupe des champions*), ne contient pas un gérondif suivi d'un complément (« فوزه بكأس الأبطال ») ce qui le caractérise par une lecture verbale plutôt qu'un simple gérondif. Cette lecture verbale est entièrement analysée.

[4] Le FTB est constitué de 12,891 phrases annotées contenant plus que 383,000 mots (Abeillé et al., 2000).

L'ATB dans sa nouvelle forme, après les améliorations et les corrections effectuées permettant d'aligner étroitement son annotation aux catégories de la grammaire arabe traditionnelle, représente une source assez robuste offrant plusieurs spécificités servant à la réalisation d'une induction de GP réussie. Ces spécificités sont citées dans la section suivante.

3 Etude de l'ATB

Avant d'expliquer les spécificités liées à l'ATB, une brève présentation de cette ressource linguistique s'avère nécessaire. En effet, l'ATB a été construit dans le cadre d'un projet en 2001 au LDC[5] (Maamouri et Bies 2004). Il représente un corpus composé de 23,611 phrases extraites d'articles de presse annotées manuellement. Ce corpus a été divisé en ensembles de textes (divisions) pour répondre aux besoins de recherche variés dans le domaine de TALN comme l'apprentissage et l'évaluation (Diab et al., 2013).

Doté d'une annotation très riche, l'ATB se caractérise par un ensemble de particularités et de qualités servant à une meilleure induction de GP. En effet, ses annotations présentent l'avantage d'être fiables. Ceci est prouvé par son efficacité dans un grand nombre de travaux dans différents domaines de TAL (Habash, 2010). Son texte source a prouvé également sa pertinence par son exploitation pour la création d'autres treebanks arabes comme le PADT (Hajic et al., 2001) et le CATiB (Habash et Roth, 2009) qui ont converti l'ATB vers leurs représentations syntaxiques en plus d'autres textes qu'ils ont annotés. De plus, l'ATB est disponible en cinq formats différents (voir la sous-section 3.1). Une autre particularité qui peut être remarquée est celle de la granularité forte qui caractérise son annotation (voir la sous-section 3.2). Finalement, l'ATB a prouvé son aptitude à représenter correctement certains phénomènes particuliers de la langue arabe (voir la sous-section 3.3).

3.1 Les formats de représentation des données dans l'ATB

L'ATB est fourni sous différents formats pour étendre son utilisation pour différents besoins de recherche. Ces formats que nous citons sont au nombre de cinq. Le format *« sgm »* représente les documents sources. Par contre, le format *« pos »* affiche des informations (comme la translitération, la voyéllation et la traduction) décrivant chaque mot source sous forme de champs avant et après la séparation des agglutinations. Le format *« xml »* quant à lui affiche les annotations de l'arbre de mots sources après la séparation des agglutinations. Le format *« penntree »* représente le corpus en deux versions (voyellée ou non) sous forme d'arborescence affichant chaque mot dans sa structure hiérarchique et devant son étiquette POS. Finalement, le format *« integrated »* affiche des informations aussi bien sur la structure arborescente que sur chaque mot source avant et après la séparation des agglutinations.

Après la présentation de ces différents formats, il faut prendre une décision concernant le choix du format de l'ATB à utiliser pour induire la CFG qui sera l'entrée de l'étape d'induction de la GP. Ce choix dépend de trois critères à prendre en compte à savoir : la simplicité de représentation, la présence d'une structure arborescente et l'annotation du niveau syntaxique des documents sources. Nous avons défini ces critères en nous appuyant sur la forme de la CFG à induire. Cette grammaire se limite au niveau des étiquettes POS et non pas au niveau des mots sources. Le format « penntree » a été le seul sélectionné pour sa satisfaction à tous les critères de choix indiqués. Plus particulièrement, nous avons utilisé la version voyellée du format « penntree » pour éviter les ambigüités liées à l'absence de voyelles en arabe.

3.2 Les niveaux de granularité des catégories

L'annotation dans l'ATB est caractérisée par une granularité forte. En effet, cette annotation inclut plus que 400 étiquettes POS différentes offrant des informations morphosyntaxiques comme la déclinaison, le mode, le genre, la définition, etc (Maamouri et al., 2009). Parmi ces étiquettes, 22 sont syntagmatiques (des catégories syntaxiques), 20 sont des relations syntaxiques et sémantiques et 24 représentent les étiquettes POS de base. L'ATB prend en compte également des pronoms vides qui peuvent apparaitre dans les phrases arabes tout en leur affectant une étiquette spécifique. Cette annotation a été toujours améliorée pour résoudre les incohérences morphosyntaxiques liées à certaines spécificités de la langue arabe (Maamouri et al., 2008 ; Kulick et al., 2010). La figure 1 illustre les différents traits caractérisant la plupart des catégories lexicales décrites dans l'ATB (Maamouri et al., 2009).

Nature	**Nom**	**Verbe**	**Verbe au présent**	**Pronom**	**Pronom Relatif**

[5] LDC (Linguistic Data Consotium) : Consortium de données linguistiques https://www.ldc.upenn.edu/

Traits		ou Adjectif				
Type		Nom : NUM, PROP, QUANT, VN Adjectif : COMP, NUM, VN	I, C, P	---	POSS, REL, DEM	---
Fonction		---	SUBJ	---	---	---
Déclinaison (mode)		NOM, ACC, GEN	---	I, JUS, SJ	---	NOM, ACC, GEN
Définition		DEF, INDEF	---	---	---	DEF, INDEF
Déterminant		DET,	---	---	---	---
Forme		---	PASS,	---	---	---
Accord	Genre	MASC, FEM	M, F	---	M, F	---
	Nombre	SG DU PL	S, D, P	---	S, D, P	
	Personne	---	1, 2, 3	---	1, 2, 3	
Exemples		NOUN_NUM+NSUFF_FEM_SG+CASE_INDEF_NOM	CV+CVSUFF_SUBJ:2MP	IV1P+IV_PASS+IVSUFF_MOOD:I	POSS_PRON_2MP	REL_PRON+CASE_DEF_GEN
		DET+ADJ_COMP+NSUFF_MASC_DU_ACC	PV_PASS+PVSUFF_SUBJ:3MD	IV3MP+IV+IVSUFF_SUBJ:MP_MOOD:SJ	DEM_PRON_MD	REL_PRON+CASE_INDEF_ACC

FIGURE 1 : Les traits caractérisant des catégories lexicales de l'ATB

Maintenant si nous diminuons cette granularité, plusieurs sous-ensembles de catégories seront factorisés en une seule catégorie. Prenons comme exemple, le sous-ensemble {NOUN_PROP, NOUN_PROP+CASE_DEF_ACC, NOUN_PROP+CASE_DEF_NOM} qui marque les noms propres par trois étiquettes dans le cas d'une forte granularité. Si nous avons une faible granularité, ces étiquettes sont généralisées, et factorisées en une seule étiquette sous le nom NOUN_PROP. Ceci s'explique par le fait que le manque de précision dans les catégories grammaticales dû à la diminution du niveau de granularité permet de les factoriser. Cette factorisation influence le nombre de règles de la CFG, et par conséquent sa taille. En effet, plus ce niveau est bas, plus le nombre de catégories grammaticales est réduit et plus la CFG est compacte mais abstraite et générale. Inversement, plus ce niveau est élevé, plus le nombre de catégories est grand et plus la CFG est détaillée mais précise et significative. La dégradation de la significativité dans la CFG est due à la perte d'informations au niveau des règles lorsqu'on réduit la granularité de ses catégories. Il faut alors contrôler le niveau de granularité des catégories pour pouvoir faire un compromis entre la taille et la qualité de la CFG.

3.3 Représentation de certains phénomènes particuliers de la langue arabe

Comme nous l'avons déjà mentionné dans l'introduction de cet article, la langue arabe présente plusieurs défis lors de son traitement automatique. Parmi ces défis, nous citons les phénomènes linguistiques particuliers comme les relatives, les coordinations et les anaphores. L'ATB a aussi contribué à traiter ces phénomènes en les représentant conformément à la grammaire arabe (Maamouri et al., 2009), ce qui favorise la robustesse des ressources qui peuvent l'exploiter.

Dans le cas des propositions relatives, il est bien remarquable, comme le montre la figure 2, que la relative SBAR « التي لم تحترق» (Al~atiy lam taHotariq+o/ *qui ne sont pas brulées*) est réellement jointe au syntagme nominal « المواد الهيدروكربونية» (Al+mawAd~+i Al+hiydoruwkarobuwniy~ap+i/ *les matières hydro-carboniques*) qui la modifie.

```
(NP (NP -Al+mawAd~+i::-□ المَوادِّ ::the+substances/materials+[def.gen.]
        Alhydrwkrbwnyp:: الهيدروكربونية ::nogloss )
    (SBAR (WHNP-2 Al~atiy:: الَّتي ::which/who/whom_[fem.sg.] )
          (S (VP (PRT lam:: لَم ::did_not )
                 ta+Hotariq+o:: تَحْتَرِقْ ::it/they/she+burn_up/be_burned+[jus.]
                 (NP-SBJ-2 *T*)))))
```

FIGURE 2 : Un exemple de proposition relative représenté par l'ATB (Maamouri et al., 2009)

Pour les coordinations, elles sont composées en arabe généralement de deux conjoints ainsi que la conjonction qui les réunit. L'ATB spécifie plusieurs formes de coordinations selon la manière de représentation des trois composants de la coordination. Concernant les anaphores, l'ATB indique uniquement ceux des catégories vides et des cas exceptionnels comme les structures écartant les syntagmes verbaux (Maamouri et al., 2009).
La richesse et la fiabilité des annotations de l'ATB, ainsi que la pertinence de ses documents sources nous ont incités à l'utiliser pour générer automatiquement une GP héritant les qualités de ce treebank. Ceci permet de favoriser sa robustesse tout en gagnant en temps de construction. Nous proposons alors une démarche à appliquer pour exploiter cette ressource et obtenir notre GP.

4 Le formalisme de GP

Le formalisme de GP représente une approche s'appuyant sur les contraintes (Blache, 2001) tout en permettant un accès direct aux valeurs des variables. En effet, il représente les informations syntaxiques directement en fonction de catégories, et non pas en fonction de structures comme le font les autres approches basées sur la satisfaction de contraintes (Pollard, 1994 ; Maruyama, 1990). Le formalisme de GP s'inscrit dans le cadre des grammaires syntagmatiques tout en adoptant une structure syntaxique hiérarchisée. Ainsi, une GP est formée par un ensemble de propriétés exprimant différentes relations entre les catégories qui forment la structure syntaxique. Dans ce qui suit, nous présentons ses éléments essentiels ainsi que son fonctionnement.

4.1 Les catégories

Une catégorie en GP est formée par une structure de traits définissant les informations pouvant intervenir dans la spécification de contraintes. Chaque trait est un couple <étiquette, valeur>. Les traits partageant des caractéristiques communes sont regroupés dans un type spécifique. A ce type, plusieurs sous-types peuvent être associés héritant ses traits, ainsi que des traits spécifiques. Les types et leurs sous-types peuvent être organisés sous la forme de hiérarchies, distinguant ainsi différents niveaux de spécification de l'information, de telle sorte que chaque catégorie ayant un ensemble de traits est associée à un certain niveau de spécification (granularité) de son type. Une hiérarchie est représentée sous la forme d'un arbre où la racine représente un type et les nœuds descendants sont des sous-types plus spécifiques de leurs nœud parent. La figure 3 ci-dessous illustre la hiérarchie du type « *cat_lexicale* » caractérisant les catégories lexicales de l'ATB. Pour ce type, un seul trait appelé CATLEX est spécifié ayant une valeur catlex complexe dont le premier trait est appelé NATURE. Ces traits sont représentés dans une matrice sous le type concerné, leurs étiquettes sont en majuscules et leur type est en italique.

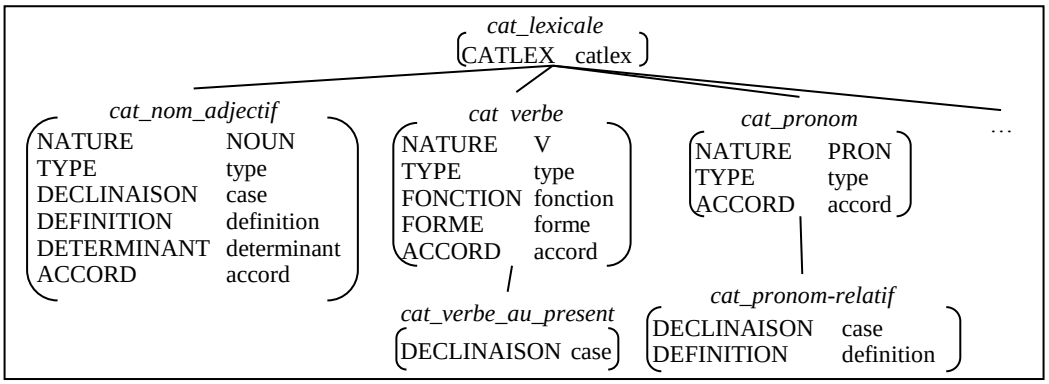

FIGURE 3 : Hiérarchie du type « *cat_lexicale* » caractérisant les catégories lexicales dans l'ATB

4.2 Les Propriétés

Une propriété est une contrainte portant sur un ensemble de catégories et décrivant une certaine catégorie. Ce qui caractérise les propriétés est le fait qu'elles sont toutes définies au même niveau, c'est à dire qu'elles ne sont ni dépendantes les unes des autres ni ordonnées entre elles. En plus, elles représentent des relations traitant toutes les informations de manière explicite à la différence des représentations de constituants qui se limitent à la définition explicite d'une relation unique : la hiérarchie. Les relations hiérarchiques représentent l'information syntaxique de manière holistique. Cette représentation ignore les cas où cette information est incomplète ou mal formée. Ce type de relations ne traite pas les phénomènes linguistiques complexes comme les relatives, les anaphores et les coordinations. Les propriétés par contre peuvent décrire ces phénomènes grâce à leur représentation décentralisée et locale des informations linguistiques. Ces propriétés peuvent être du niveau lexical (comme les propriétés morphologiques ou phonologiques) ou bien du niveau syntaxique. Les propriétés syntaxiques portent sur six différents types de contraintes montrées dans la figure 4 suivante :

Propriétés	Symboles	Fonctions
Linéarité (Lin)	≺	Relations de précédence linéaire entre les constituants d'un constituant d'un niveau syntaxique
Unicité (Unic)	Unic	Ensemble des constituants ne devant apparaître qu'une seule fois

Obligation (Oblig)	Oblig	Ensemble des têtes possibles du constituant de niveau syntaxique
Exigence (Exig)	⇒	Cooccurrence obligatoire entre les constituants
Exclusion (Excl)	⊗	Restriction de cooccurrence entre les constituants
Dépendance (Dep)	∾	Relations de dépendance entre les constituants

FIGURE 4 : Fonctions des propriétés dans les GP

Les propriétés d'unicité et d'obligation sont des relations unaires. Les autres sont par contre des relations binaires.

4.3 Vérification de la satisfaction de contraintes

Le formalisme de GP représente les contraintes de manière indépendante. Ces contraintes sont regroupées dans des sous-systèmes caractérisant chacun une catégorie syntaxique. L'analyse avec ce formalisme revient en fait à vérifier pour chaque catégorie syntaxique la satisfaisabilité de son sous-système de contraintes. Pour analyser un énoncé donné, un processus de trois étapes est à appliquer : L'énumération de l'ensemble des catégories possibles de cet énoncé y compris celles syntaxiques susceptibles d'être des catégories mères pour les catégories dégagées, la construction des suites possibles des catégories énumérées, et finalement le calcul de la caractérisation des suites par la vérification de la consistance des sous-systèmes de contraintes correspondant aux catégories syntaxiques de ces suites. L'apport du formalisme de GP est très bien décrit grâce à cette notion de caractérisation. En effet, aucune règle syntagmatique ni schéma de règle n'est nécessaire pour décrire syntaxiquement un énoncé. Il suffit de fournir un ensemble de systèmes de contraintes décrivant cet énoncé de façon simple et directe et peu importe sa forme, et de vérifier leur satisfaction.

5 Démarche proposée

Pour élaborer notre démarche, nous nous sommes basées sur l'idée d'induction de GP à partir du FTB adoptée dans (Blache et Rauzy, 2012). Cette démarche s'articule autour de deux grandes étapes : l'induction de la CFG et l'induction de la GP. La démarche proposée est présentée dans la figure 5 suivante :

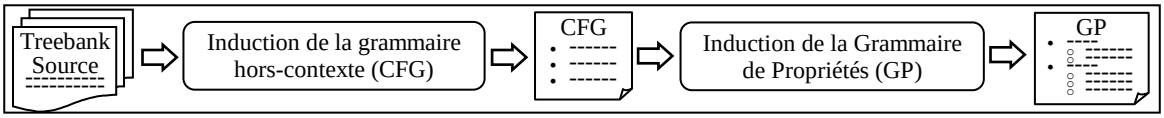

FIGURE 5 : Démarche d'induction de la GP

5.1 Etape d'induction de la CFG

L'induction de notre GP ne peut pas être réalisée directement en appliquant une simple tâche d'acquisition et de manipulation des données du treebank. Il faut en effet introduire une étape intermédiaire permettant de représenter les productions décrivant les unités syntaxiques. Cette étape consiste à parcourir l'ATB et à en extraire les constructions (règles) possibles pour produire une CFG pour l'arabe à différents niveaux de granularité. Ceci est réalisé dans le cadre de trois sous-étapes primordiales : la détection des constituants, l'élaboration des règles et le contrôle de ses niveaux de granularité (voir figure 6).

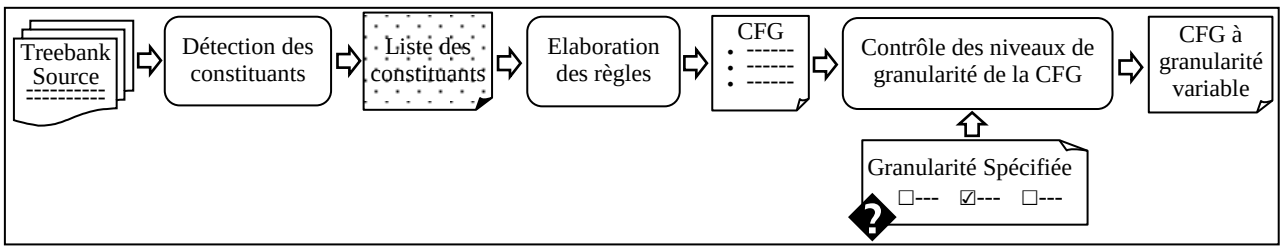

FIGURE 6 : Sous-étapes d'induction de la CFG

5.1.1 Détection des constituants

Cette sous-étape permet tout d'abord de parcourir le treebank ligne par ligne en décomposant chaque ligne en un ensemble de mots. Les mots commençant par une parenthèse ouvrante représentent les constituants. Après l'acquisition de l'ensemble des constituants, elle élimine les doublons puis trie ces constituants par ordre alphabétique. Finalement, afin de présenter quelques statistiques, cette sous-étape calcule le nombre d'occurrences de chacun de ces constituants.

5.1.2 Elaboration des règles

Il s'agit d'extraire les différentes constructions possibles (règles) pouvant être représentées dans l'ATB, ce qui permet de produire implicitement la CFG. Ceci se fait en parcourant ligne par ligne les fichiers des données source du treebank. Pour chaque ligne, trois tâches sont réalisées successivement à savoir : la récupération de l'ensemble des mots formant cette ligne, la détection des extrémités des règles en utilisant les mots récupérés et la génération des règles. La deuxième tâche permet d'affecter des positions incrémentales aux mots récupérés tout en donnant aux bornes (les parenthèses ouvrante et fermante) de chaque règle détectée la même position. A la troisième tâche, il s'agit de former les règles à générer. Chaque règle est composée d'une partie gauche consacrée à l'un des constituants du niveau syntaxique, et d'une partie droite représentant les constituants descendants de ce constituant syntaxique. Cette tâche s'étend également à l'exploration des constituants descendants dans la partie droite à la recherche d'autres règles de niveaux hiérarchiques plus fins. Après chaque parcours d'un fichier courant, les règles extraites sont accumulées à la liste des règles extraites des fichiers déjà parcourus. Cette liste est filtrée et triée. La liste finale des règles forme notre CFG. Il est possible également de calculer le nombre d'occurrences de chaque règle dans la liste pour présenter quelques statistiques.

En se trouvant devant une complexité prévue de la forte granularité caractérisant l'annotation de l'ATB, les catégories ne doivent pas être détectées d'emblée, mais le contrôle de leurs niveaux de granularité s'impose. L'emploi de ce contrôle est également motivé par le fait que nous voulons générer une grammaire à large couverture. Le fait d'offrir une certaine souplesse de présentation à cette grande masse de données favorise la qualité de la grammaire obtenue. Le contrôle à appliquer s'effectue au niveau de l'étape d'induction de la CFG vu qu'il concerne les catégories détectées. La section suivante explique les mécanismes de ce contrôle.

5.1.3 Contrôle des niveaux de granularité de la CFG

Pour pouvoir contrôler la granularité des catégories détectées, il faut spécifier leurs traits. Ces traits peuvent être organisés selon le formalisme de hiérarchies des types (voir la section 4.1). Mais, il faut les extraire tout d'abord de la structure de leur catégorie concernée. Pour cela, nous avons effectué une étude de cette structuration, et constaté que l'ATB a utilisé huit informations dans son annotation, portant sur : la numérotation, la déclinaison, le genre et le nombre, le mode de conjugaison du verbe, son type, le déterminant, la catégorisation lexicale et la celle syntaxique. Chacune de ces informations peut être spécifiée en se basant sur des caractères de séparation comme « : », « - », « + », « _ » ainsi que des mots clés comme « NSUFF », « CVSUFF », « IVSUFF », « PVSUFF », « CASE » et « MOOD ».

5.2 Etape d'induction de la GP

La CFG étant l'entrée de cette étape et la sortie de celle précédente, est utilisée pour induire automatiquement une GP. Ce qui facilite cette étape est le fait qu'aussi bien la CFG que la GP sont structurées sous forme de constituants de niveau syntaxique auxquels nous affectons des informations de différents types. La différence réside au niveau du type de ces informations. En effet, la CFG donne pour chaque constituant syntaxique, l'ensemble de règles qui le décrit. Ces règles forment en fait des contraintes hiérarchiques. La GP, par contre, représente le constituant syntaxique à l'aide de contraintes non hiérarchiques qui sont les « propriétés ». Pour expliquer comment nous avons induit notre GP, nous nous mettons d'accord sur quelques notations :

- XP représente tout constituant syntaxique décrit dans la grammaire.
- RHS(XP) est l'ensemble des règles spécifiant chaque XP.
- const(XP) est l'ensemble sans doublons des constituants pouvant former XP. Il est obtenu en parcourant RHS(XP) pour récupérer tout constituant faisant partie d'une règle de RHS(XP), il va servir à la représentation des propriétés. Nous avons commencé par la description des cinq premiers types de propriétés. Tandis que la description des propriétés de dépendance, elle n'est pas intégrée dans le présent article. Nous présentons, dans ce qui suit, les descriptions formelles de ces types de propriétés tout en nous appuyant sur celles établies dans (Blache, 2012) :

– La linéarité (lin) : elle vérifie dans tout l'ensemble RHS(XP) la validité de chaque relation de précédence entre chaque constituant de const(XP) et un autre. Pour cela, pour chaque couple de constituants dans const(XP), nous allons considérer que cette relation est vraie tant qu'il n'existe pas un contre exemple.

$\forall (c_i, c_j) \in const(XP) \mid c_i \neq c_j$
  $\forall rhs_a \in RHS(XP)$
    Si $((\exists (c_m, c_n) \in rhs_a \mid c_m = c_i \wedge c_n = c_j)$
    Et $(\nexists (c_m, c_n) \in RHS \mid c_m = c_i \wedge c_n = c_j \wedge c_n \prec c_m))$
    alors ajouter $lin(c_i, c_j)$

– L'unicité (unic) : elle vérifie dans tout l'ensemble RHS(XP) pour chaque constituant de const(XP) s'il n'est pas répété dans la même construction de RHS(XP). Selon cette interprétation, nous supposons que le constituant à traiter est unique tant qu'il n'existe pas un cas contraire.

$\forall c_i \in const(XP)$
  card : 0
  $\forall rhs_a \in RHS(XP)$
    $\forall c_j \in rhs_a$
      Si $(c_j = c_i)$ **alors** card ← card+1
  Si (card =1) **alors** ajouter $unic(c_i)$

– L'obligation (oblig) : elle représente l'ensemble des constituants obligatoires pour former XP. Un constituant obligatoire (tête) est un constituant devant apparaitre au moins une fois dans chacune des constructions de RHS(XP).

$\forall c_i \in const(XP)$
Si $(\forall rhs_b \in RHS(XP) \mid \exists c_j \in rhs_b \wedge c_j = c_i)$ **alors** ajouter $oblig(c_i)$

– L'exigence (Exig) : elle vérifie dans tout l'ensemble RHS(XP) la validité de chaque relation de cooccurrence entre chaque couple de constituants de const(XP). Une catégorie est co-occurrente avec une autre si l'apparition de la première implique l'apparition de la deuxième dans la même construction. Cette relation n'est pas symétrique du fait que si la deuxième catégorie apparait dans une construction sans la première, cette relation est considérée valide.

$\forall (c_i, c_j) \in const(XP) \mid c_i \neq c_j$
  bool ←vrai
  $\forall rhs_a \in RHS(XP)$
    bool ← $(\exists c_n \in rhs_a \mid c_n = c_i) \wedge (\nexists c_m \in rhs_a \mid c_m = c_j)$
Si bool **alors** ajouter $exig(c_i, c_j)$

– L'exclusion (excl) : elle vérifie dans tout l'ensemble RHS(XP) la validité de chaque relation de restriction de cooccurrence entre chaque couple de constituants de const(XP). Une catégorie n'est pas co-occurrente avec une autre s'il s'est arrivé que l'une apparait avec l'autre dans une même construction. Cette relation est totalement contraire à la relation d'exigence. Et même, elle est symétrique du fait que pour que cette relation soit valide, l'apparition de l'une de ses deux catégories empêche l'apparition de l'autre.

$\forall (c_i, c_j) \in const(XP) \mid c_i \neq c_j$
  bool ← faux
  $\forall rhs_a \in RHS(XP)$
    bool ← $(\exists (c_m, c_n) \in rhs_a \mid c_m = c_i \wedge c_n = c_j)$
Si non bool **alors** ajouter $excl(c_i, c_j)$

6 Expérimentations et résultats

Nous avons utilisé comme ressources pour induire notre GP, la deuxième division avec sa version 3.1 (ATB2 v3.1), composée de 501 articles de presse contenant 144.199 segments avant la fragmentation des clitiques. Comme nous l'avons déjà mentionné, le mécanisme d'induction de la GP se déroule sur deux tâches successives qui permettent d'obtenir successivement deux types de grammaires sous le format XML : La CFG et la GP. La taille de ces grammaires dépend du niveau de granularité des catégories qu'elles décrivent, puisque toute catégorie peut être caractérisée par différents traits morphologiques. Plus le niveau de granularité des catégories est élevé, plus ces grammaires sont complexes mais leurs propriétés de plus en plus fidèles à la langue, et inversement. La CFG obtenue est composée d'ensembles des règles décrivant chaque catégorie non terminale XP. Chaque règle prend la forme d'une liste ordonnée de catégories grammaticales représentant une catégorie syntaxique XP. La table 1 montre des informations sur la CFG

obtenue au niveau de granularité le plus élevé. Cette table affiche en particulier la fréquence de la règle « PREP NP » lorsqu'elle décrit la catégorie PP (Prepositional Phrase) y compris ses sous-catégories (e.g. PP-MNR et PP-TMP), qui intègrent plus de détail (Maamouri et al., 2009). A ce niveau, il y a 263 règles de différents types. Selon ce que nous avons observé dans la table1, nous pouvons noter que la granularité la plus élevée ne fait pas une grande différence pour certaines sous-catégories de PP. Par exemple, dans la plupart des cas, la construction « PREP NP » reste la plus fréquente quel que soit la sous-catégorie de PP qu'elle décrit. Les autres règles ne sont pas fréquentes, elles partagent ensembles le reste des occurrences. La sous-catégorie PP-LOC représente un exemple. En plus de la règle « PREP NP », elle est décrite par d'autres règles que chacune d'elles ne dépasse pas les 10 occurrences et qu'elles apportent ensembles uniquement 19 occurrences. Par ailleurs, le signe "#" affecté à quelques paramètres signifie leur cardinalité.

Catégories syntaxiques	PP	PP-CLR	PP-PRP	PP-TMP	PP-LOC	PP-PRD	PP-MNR	PP-DIR	Others
Σ# Règles	50	44	15	15	13	13	12	9	--
#Occ de « PREP NP »	12834	3025	445	754	1511	762	246	154	--
Σ#Occ des règles	13814	3781	684	805	1537	805	286	165	222

TABLE 1 : Fréquence de la règle « PREP NP » décrivant les sous-catégories de PP au niveau plus haut de granularité

Mais si nous observons la CFG, illustrée en partie dans la table 2, nous pouvons noter qu'indépendamment du niveau de granularité élevé des catégories syntaxiques, les occurrences de la construction « PREP NP » représentent en tout environ 90% des cas, ce qui rend l'augmentation de la granularité des catégories inutiles dans certains cas. La réduction de cette granularité à 0 nous donne une CFG pour les PP plus compacte. Elle est illustrée en partie par la table 2 et regroupe uniquement 59 types différents de règles intégrant dans la plupart des cas la construction « PREP NP ». Nous pouvons remarquer aussi que, dans les deux CFG, les constructions de PP les plus fréquentes dans le treebank sont formées de deux constituants. Par contre, les constructions complexes formées de plus que deux constituants sont rares. Généralement, nous avons trouvé que, pour toutes les catégories non terminales, le niveau de granularité des catégories affecte également la taille de toute la grammaire. En effet, le nombre de règles dans la CFG s'est divisé par 6 au niveau le plus bas par rapport à celui le plus élevé (2998/14452).

Règles	#Occ	Règles	#Occ	Règles	#Occ
PREP NP	19886	PREP ADVP	32	PP PREP NP	10
PREP SBAR	1346	NP PREP NP	28	PREP NP PUNC	10
PREP S	237	PREP PUNC NP	22	PREP UCP	8
PP CONJ PP	126	PP PP	20	PUNC PREP SBAR PUNC	7
-NONE-	87	PUNC PREP NP	20	PREP NP PP	6
PRT PREP NP	63	ADVP PREP NP	19	14 règles	≤5
PP PUNC CONJ PP	48	PREP PUNC NP PUNC	18	25 autres règles	1
PUNC PREP NP PUNC	42	Σ# Occurrences			22099

TABLE 2 : Extrait de la CFG au plus bas niveau de granularité décrivant la catégorie PP

La GP obtenue à un niveau donné décrit pour chaque catégorie syntaxique, l'ensemble de ses constituants ainsi que les propriétés qui relient ces constituants. La figure 7 et la figure 8 illustrent respectivement des extraits des GP obtenues aussi bien au niveau de granularité le plus élevé que celui le plus bas pour la catégorie PP. Premièrement, nous pouvons remarquer que, grâce au formalisme de GP, des informations implicites de différents types dans le treebank sont rendues explicites. Ce sont les propriétés (ou relations) qui relient les différents constituants. Ces nouvelles informations peuvent servir à la tâche d'analyse syntaxique. A partir de la figure 7, prenons comme exemple la propriété de linéarité « PREP $\prec$ S-NOM » décrivant la sous-catégorie PP-DIR désignant un syntagme prépositionnel de direction. Cette relation indique que si la catégorie PREP (préposition) apparaît avec la catégorie S-NOM (proposition nominative) dans la même réalisation, elle va toujours précéder S-NOM directement ou indirectement. Une information de ce type n'est pas explicite dans le treebank. Toutefois, avec un niveau de granularité totalement élevé des catégories, plusieurs informations implicites peuvent être répétées pour plusieurs sous-catégories, ce qui multiplie la taille de la GP et rend son parcours plus difficile. Plus particulièrement dans la figure 7, c'est le cas des propriétés reliant les catégories PREP et NP. Voyons ici que ces propriétés sont répétées au moins 6 fois dans la grammaire pour les sous-catégories indiquées.

PP-DIR	Const	{PP, PREP, NP, ADVP, S-NOM, SBAR, PRT}
	Unic	{PREP, NP, ADVP, S-NOM, SBAR, PRT}
	Lin	PP $\prec$ {PREP, NP} ; PRT $\prec$ {PREP, NP} ; PREP $\prec$ {NP, ADVP, S-NOM, SBAR}
	Exig	{NP, ADVP, S-NOM, SBAR} $\Rightarrow$ PREP; PRT $\Rightarrow$ {PREP, NP}
	Excl	PP $\otimes$ {ADVP, S-NOM, SBAR, PRT}; NP $\otimes$ { ADVP, S-NOM, SBAR} ADVP $\otimes$ {S-NOM, SBAR, PRT};S-NOM $\otimes$ {SBAR, PRT};SBAR $\otimes$ {PRT}

PP-DTV PP-TPC PP-LOC PP-MNR PP-PRD	Const	{PREP, NP}
	Unic	{PREP, NP}
	Oblig	{PREP, NP}
	Lin	PREP $\prec$ NP
	Exig	NP $\Rightarrow$ PREP PREP $\Rightarrow$ NP

FIGURE 7 : Extrait de la CFG au plus haut niveau de granularité

L'induction de la GP au niveau de granularité le plus bas montre une grande différence. La figure 8 montre un extrait de cette grammaire pour la catégorie PP. La GP devient beaucoup plus compacte, les catégories sont plus simples et les propriétés ne sont pas répétées. C'est parce que ces catégories ont été généralisées et factorisées Toutefois, ce manque de précision peut perdre l'information. Plusieurs exemples dans la GP peuvent prouver cette idée : Ainsi, avant la généralisation des catégories, la propriété de linéarité «PRON_3MS ≺ DET+ADJ+CASE_DEF_NOM» décrit la sous-catégorie NP-ADV-1. Après la généralisation, il faut que la précision de cette propriété soit réduite, et la propriété soit transformée à une autre étant « PRON ≺ ADJ » pour décrire la catégorie de base NP. Mais ceci n'a pas été effectué. Ceci s'explique par le fait que la validité de cette propriété n'a pas été garantie pour toutes les sous-catégories de NP. L'absence de plusieurs propriétés à cause de la généralisation peut produire un degré d'erreur.

Const	{-NONE-, NP, S, SBAR, PP, PREP, ADVP, PRT, CONJP, UCP, NAC, FRAG, PRON, TYPO}
Unic	{-NONE-, PREP, NP, ADVP, SBAR, PRT, PRON, UCP, NAC, FRAG}
Lin	-NONE- ≺ NP ; NP ≺ {UCP, NAC, FRAG ; PRT ≺ {PREP, NP, S, PRON, PP, SBAR, ADVP} ; PP ≺ {PREP, NP, S, NAC}; TYPO ≺ {PP, S}; PREP ≺ {NP, ADVP, S, UCP, PRON}; UCP ≺ PP
Exig	{NP, ADVP, S, SBAR} ⇒ PREP ; PRT ⇒ {PREP, NP};{CONJP, NAC, FRAG} ⇒NP
Excl	{TYPO, PRT, CONJP, PRON, ADVP, S, SBAR, -NONE-} ⊗ {UCP, NAC, FRAG} ; PRON ⊗ {NP, CONJP, TYPO} -NONE- ⊗ {S, SBAR, PP, ADVP, TYPO, CONJP, PRON, PRT}; NAC ⊗ FRAG; CONJP ⊗ {PRT, TYPO} S ⊗ {SBAR, PP, NP, ADVP, CONJP, PRT}; SBAR ⊗ {PP, NP, PRT, CONJP, PRON, ADVP, TYPO} PP ⊗ {ADVP, PRT, CONJP, PRON, FRAG} ; ADVP ⊗ {NP, PRON, PRT, CONJP, TYPO};PRT ⊗ TYPO

FIGURE 8 : Extrait de la GP au plus haut niveau de granularité

Dans la figure 8, nous pouvons remarquer qu'aucune propriété d'obligation n'est présentée. C'est parce que l'interprétation de ce type de propriétés exige la présence d'un constituant en une seule forme dans toutes les règles. En principe, ce constituant est la catégorie NOUN dans notre cas. Ceci n'est pas assuré, puisqu'à chaque fois, nous avons soit une catégorie plus détaillée (NOUN_NUM ou NOUN_PROP) ou bien une autre totalement différente. Tandis que les propriétés d'exclusion, elles sont très nombreuses en adoptant l'interprétation que nous venons de présenter dans la section 5.2. Considérer que toute absence de deux constituants d'une catégorie syntaxique dans toutes ses constructions représente une relation de restriction de cooccurrence n'est pas toujours valide. En effet, la rareté de ces règles dans la langue ou la non richesse du treebank en entrée peuvent être les vraies raisons de cette absence.

Selon les résultats obtenus, nous constatons que la variation du niveau de granularité des catégories a une grande influence sur la réduction de la complexité du problème permettant ainsi de diminuer la taille de la GP induite. Mais, même la généralisation pose un problème du faite qu'elle engendre une perte dans la précision de l'information. Ceci favorise beaucoup plus l'adoption d'un mécanisme de contrôle du niveau de granularité pour faire le compromis entre la généralisation et la spécification des catégories grammaticales de l'ATB.

7 Conclusions et perspectives

Nous avons proposé dans cet article une démarche de construction d'une GP à granularité variable à partir de l'ATB, ce qui la rend une ressource à large couverture héritant des qualités de l'ATB comme sa fiabilité, sa soumission à des consensus et sa richesse en annotations de différents types. La technique que nous avons adoptée pour construire cette ressource présente l'avantage d'être générique. En effet, elle est indépendante non seulement de toute langue, mais aussi, du formalisme source, puisque la génération des propriétés se fait directement à partir de la CFG. En plus, avec les types de propriétés définis jusqu'à présent, cette technique est automatique, ce qui favorise sa réutilisabilité pour des treebanks de différentes langues et de différents formalismes sources.

Dans le but, d'offrir une représentation très précise de l'information syntaxique, l'ensemble des relations présentées dans la GP peut toujours être enrichi ou modifié. Cette grammaire peut être également utilisée pour enrichir le treebank arabe ATB, qui est à base de syntagmes, avec la représentation à base de propriétés, ce qui peut permettre d'améliorer la qualité du treebank. Pour optimiser cet enrichissement, plusieurs mécanismes de contrôle peuvent être intégrés au niveau de la détermination des unités syntaxiques, et de l'efficacité de leurs propriétés linguistiques.

Références

ABEILLÉ A., CLÉMENT L., KINYON A. (2000). Building a treebank for French. Proceedings of *the Second International Language Resources and Evaluation Conference*. Athens, Greece.

AURAN C., BOUZON C., HIRST D.J. (2004). The Aix-MARSEC project: an evolutive database of spoken British English. Proceedings of *the Second International Conference on Speech Prosody*, 561-564. Nara.

BLACHE P. (2001). *Les Grammaires de Propriétés : Des contraintes pour le traitement automatique des langues naturelles*. Hermès Sciences Publications.

BLACHE P. (2014). A Chinese Constraint Grammar Extracted from the Chinese Treebank. Proceedings of *APCLC*.

BLACHE P., GUÉNOT M.-L. & VANRULLEN T. (2003). Corpus-based grammar development. Proceedings of *Corpus Linguistics-03*.

BLACHE P., RAUZY S. (2012). Hybridization and Treebank Enrichment with Constraint-Based Representations. Proceedings of *LREC-2012*.

DUCHIER D., PROST J.-P., DAO T.-B-.H. (2009). A Model-Theoretic Framework for Grammaticality Judgements. FG, volume 5591 of *Lecture Notes in Computer Science*, page 17-30. Springer.

KAY P., FILLMORE C. (1999).Grammatical Constructions and Linguistic Generalizations: the what's x doing y construction. *Language.*

LAMMIE GLEEN M., STRASSEL S. (2005). Linguistic Resources for Meeting Speech Recognition. *MLMI-05*.

MAAMOURI M., BIES A. (2004). Developing an Arabic Treebank: Methods, Guidelines, Procedures, and Tools. Proceedings of *COLING-04*. Geneva, Switzerland.

MAAMOURI M., BIES A., KROUNA S., GADDECHE F., BOUZIRI B. (2009). Penn Arabic Treebank guidelines v4.8. *Technical report, Linguistic Data Consortium*, University of Pennsylvania.

MAAMOURI M., ZAGHOUANI W., VIOLETTA CAVALLI-SFORZA V., GRAFF D., CIUL M. (2012). Developing ARET: An NLP-based Educational Tool Set for Arabic Reading Enhancement. *NAACL-HLT 2012*. Montreal.

MOHRI M., ROARK B. (2006). Probabilistic Context-Free Grammar Induction Based on Structural Zeros. Proceedings of *the Seventh Meeting of the Human Language Technology conference- North American Chapter of the Association for Computational Linguistics (HLT-NAACL 2006)*. New York.

REHBEIN I., VAN GENABITH. J. (2007) Treebank annotation schemes and parser evaluation for German. Proceedings of *the Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, pages 630–639*. Prague, Czech Re public.

SONG Z., ISMAEL S., GRIMES S., DOERMANN D., STRASSEL S. (2012). Linguistic Resources for Handwriting Recognition and Translation Evaluation. *LREC 2012*. Istanbul.

TOUNSI L., VAN GENABITH J. (2010). Arabic Parsing Using Grammar Transforms. *LREC 2010*.

VANRULLEN T., BLACHE P., PORTES C., RAUZY S., MAEYHIEUX J.-F., GUENOT M.-L., BALFOURIER J.-M., BELLENGIER E. (2005). Une plateforme pour l'acquisition, la maintenance et la validation de ressources lexicales. Actes de *TALN, pp. 41-48*. Paris, France.