Générer une grammaire d'arbres adjoints pour l'arabe à partir d'une méta-grammaire

Cherifa Ben Khelil 1, 2
(1) LIFO, Bâtiment IIIA 6 Rue Léonard de Vinci, F-45067 Orléans, France
(2) RIADI, Campus Universitaire de la Manouba, 2010 Manouba, Tunisie
cherifa.ben-khelil@etu.univ-orleans.fr cherifa.bk@gmail.com

RÉSUMÉ
La rareté des ressources numériques pour la langue arabe, telles que les grammaires et corpus, rend son traitement plus difficile que les autres langues naturelles. A ce jour il n'existe pas une grammaire formelle à large couverture de l'arabe. Dans ce papier, nous présentons une nouvelle approche qui facilite la description de l'arabe avec le formalisme des grammaires d'arbres adjoints en utilisant une méta-grammaire. Nous exposons les premiers résultats de notre grammaire ainsi que les problèmes rencontrés pour son évaluation.

ABSTRACT
Generate a tree adjoining grammar for arabic from a meta-grammar

The scarcity of digital resources for arabic language, such as grammars and corpus, makes its processing more difficult than other natural languages. To date there is not a wide-coverage grammar of the Arabic langage. In this paper, we present a new approach in order to facilitate the description of Arabic with Tree Adjoining Grammar using a meta-grammar. We present our first results as well as problems encountered in the evaluation of the grammar..

MOTS-CLÉS : Arabe, grammaire d'arbres adjoints, méta-grammaire, évaluation.

KEYWORDS: Arabic, Tree Adjoining Grammar, meta-grammar, evaluation.

1 Introduction

En la linguistique et l'informatique, une grammaire formelle constitue un moyen de représentation permettant de définir une syntaxe ou plus largement un langage formel. Elle est utilisée dans plusieurs applications dans le domaine du Traitement Automatique du Langage Naturel (TALN) (systèmes de traduction automatiques, dialogue homme-machine, etc.).
Nous nous sommes intéressés à la langue arabe qui est une langue sémitique. L'arabe présente beaucoup de spécificités à différents niveaux : phonologique, morphologique, syntaxique et aussi sémantique. Bien que plusieurs travaux de recherche aient abordé ces problématiques, les ressources numériques utiles pour le traitement de l'arabe demeurent relativement rares. En effet, la majorité de ces ressources libres traite de la morphologie (Xerox Arabic Morphological Analysis and Generation [1], ElixirFM [2], etc.) mais peu d'entre elles sont disponibles pour la syntaxe. Nous pouvons citer la grammaire générative transformationnelle (Alkhuli, 1979; Kebbe, 2000), d'autres modèles linguistiques

1. https ://open.xerox.com/Services/arabic-morphology
2. https ://github.com/otakar-smrz/elixir-fm

tels que la grammaire lexicale fonctionnelle (LFG) (Fehri, 1981) et la grammaire syntagmatique guidée par les têtes (HPSG) (Mutawa *et al.*, 2008; Haddar *et al.*, 2010) . Cependant, ces grammaires ont une couverture limitée et à ce jour il n'existe pas un analyseur syntaxique à large couverture de l'arabe. Outre les grammaires, les corpus formés de phrases analysées et stockées sous forme de banques d'arbres (Treebanks) constituent un autre type de ressources syntaxiques mais ils ne sont pas en accès libre.

C'est dans ce contexte que s'inscrit notre travail de recherche qui vise à élaborer une grammaire formelle pour l'utiliser dans les applications du traitement automatique de la langue arabe. Notre choix s'est porté sur le formalisme des grammaires d'arbres adjoints (TAG)(Joshi *et al.*, 1975). Ce choix est motivé par le pouvoir de représentation des TAG (les structures simples, complexes, combinatoires, partagées, etc...) et leur capacité de traiter certains phénomènes tels que les enchâssements. La grammaire que nous proposons est produite semi automatiquement grâce au le langage de description méta-grammatical XMG (eXtensible MetaGrammar) (Crabbé *et al.*, 2013).

Cet article est organisé de la manière suivante. Dans la section 2, nous présentons le formalisme TAG et son application à l'arabe. Dans la section 3, nous exposons notre approche qui décrit l'arabe au moyen d'une méta-grammaire. Finalement, dans la section 4, nous discuterons des problèmes rencontrés lors de l'évaluation de la qualité de la grammaire.

2 Représentation de la syntaxe de l'arabe au moyen de TAG

2.1 Présentation du formalisme TAG

La grammaire d'arbres adjoints (Joshi *et al.*, 1975) est un formalisme syntaxique qui permet de rendre compte des liens entre les constituants de la phrase. Il offre un système de réécriture d'arbres dont les unités sont des arbres élémentaires. Ceux-ci sont définis par un ensemble d'arbres initiaux et d'arbres auxiliaires.

- Un arbre initial est un arbre fini, dont les nœuds feuilles sont soit des symboles terminaux, soit des symboles non terminaux. Les symboles non terminaux sont appelés nœuds de substitution et sont marqués par le symbole (↓).
- Un arbre auxiliaire a en outre un nœud feuille étiqueté par un symbole non terminal appelé "nœud pied" et il est marqué par le symbole (*). Le nœud pied et la racine de l'arbre auxiliaire sont nécessairement de même catégorie.

Les deux opérations de réécriture d'arbres autorisées par TAG sont : l'adjonction et la substitution. A l'issue de ces opérations de réécriture, on obtient un nouvel arbre appelé arbre dérivé.

L'opération de substitution (Figure 1) permet d'insérer un arbre α à la frontière d'un arbre dérivé initial contenant un nœud de substitution β. La substitution est autorisée uniquement si le nœud de substitution et le nœud racine, respectivement de β et de α sont étiquetés par un symbole identique.

L'opération d'adjonction (Figure 2) permet, plutôt, d'insérer l'arbre γ dans l'arbre β sur un nœud interne X. Le nœud X situé dans β est remplacé par la racine de γ. L'adjonction est autorisée si la catégorie du nœud X est identique à la catégorie du nœud racine de γ.

TAG est légèrement sensible au contexte (mildly contextsensitive) (Weir, 1988). C'est un formalisme plus puissant que les grammaires hors contextes [3], mais strictement incluse dans la classe des

3. C'est une grammaire de réécriture dont les parties gauches des règles contiennent un unique non-terminal, donc sa dérivation ne dépend d'aucun contexte (X $\longrightarrow$ w)

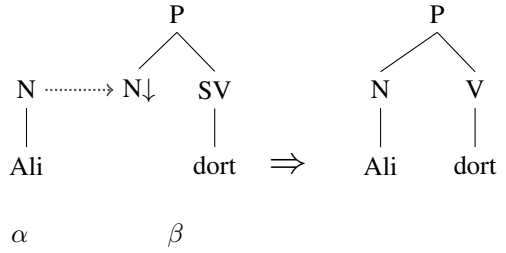

FIGURE 1 – Substitution de l'arbre α dans l'arbre β

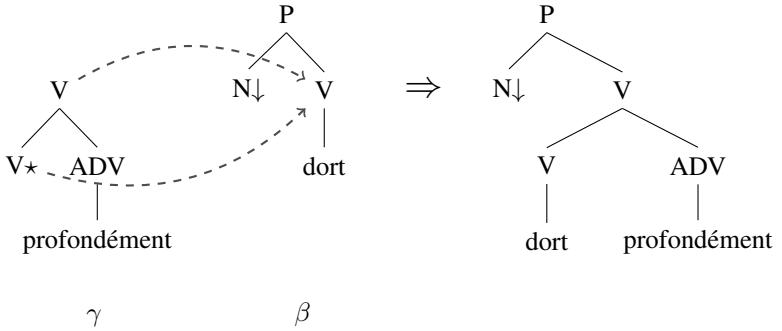

FIGURE 2 – Adjonction de l'arbre γ dans l'arbre β

grammaires contextuelles [4]. Elle définit un domaine de localité étendu étant donné que la profondeur des arbres élémentaires est variable, au contraire des règles de réécriture hors contexte dont la profondeur est égale à 1. De ce fait, elle possède un fort pouvoir génératif, qui englobe les dépendances à longue distance et certaines dépendances croisées ainsi qu'une factorisation des composantes grammaticales récursives. De plus, d'un point de vue traitement, TAG reste analysable en un temps polynomial O (n^6). Nous ne pouvons pas affirmer que ce formalisme est incontestablement le meilleur pour représenter l'arabe. Néanmoins, ses caractéristiques ont permis de représenter des phénomènes de l'arabe tels que l'enchâssement.

2.2 Capacité à gérer l'enchâssement

La représentation de ce phénomène est possible avec TAG grâce à l'opération d'adjonction.Cette opération permet d'intercaler une structure complète dans une autre structure rendant ainsi la représentation des dépendances enchâssées très naturelle. L'adjonction d'un arbre auxiliaire à lui-même ou à un autre arbre élémentaire met en valeur la récursivité de la langue naturelle.

Considérons l'exemple (Figure 3) du syntagme nominal de liaison (مركب موصولي) الذي حقق النجاح

Ce type de grammaire permet de définir les langages algébriques qui sont reconnaissables par un automate à pile non déterministe.

4. Les règles d'une grammaire contextuelle sont restreintes en obligeant le membre droit à être au moins aussi long que le membre de gauche (uXv ⟶ uwv)
Le langage engendré par ce type de grammaire est récursif et reconnaissable par un automate borné linéairement.

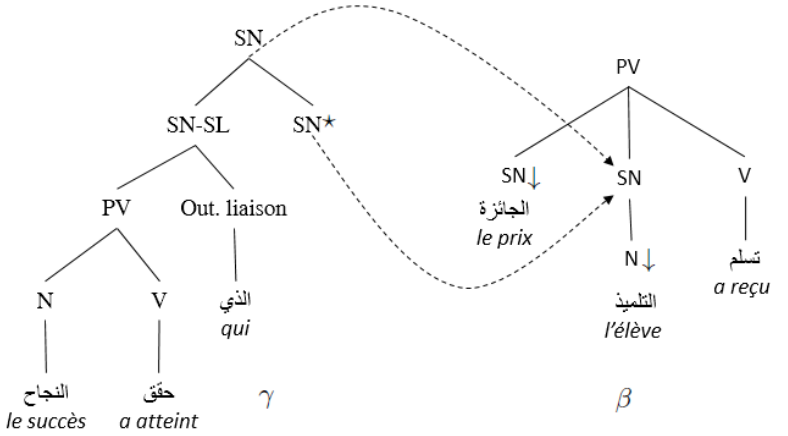

FIGURE 3 – Exemple représentatif de la manipulation des structures enchâssées en TAG

(qui a atteint le succès), représenté par l'arbre γ, il peut être enchâssé dans une phrase principale تسلم التلميذ الجائزة (l'élève a reçu le prix) dont la structure est représentée par l'arbre β, et donner ainsi naissance à une nouvelle phrase plus complexe qu'est : تسلم التلميذ الذي حقق النجاح الجائزة (l'élève qui a atteint le succès a reçu le prix).

2.3 ArabTAG V1.0 (Arabic Tree Adjoining Grammar)

A notre connaissance, il existe deux travaux qui ont construit une TAG pour l'arabe : une TAG par extraction d'arbres élémentaires à partir du corpus Penn Arabic TreeBank (Habash & Rambow, 2004) et ArabTAG (Ben Fraj, 2011). Notre travail de thèse prend ses origines de cette dernière.
ArabTAG (Arabic Tree Adjoining Grammar) hérite de tous les fondements de base de TAG.
Elle décrit différents composants syntaxiques de différents niveaux : phrases, syntagmes et mots ainsi que les différentes informations qui leur sont relatives (morphologiques et syntaxiques).
Dans le but de limiter le nombre d'arbres élémentaires, seules les structures syntaxiques, formées des éléments essentiels de la phrase, sont représentées telles que les phrases verbales (Verbe, sujet et complément d'objet), phrases nominales et les syntagmes nominaux.
De plus, pour la bonne gestion des compositions syntaxiques, (Ben Fraj, 2010) a défini différents types d'informations (morphologiques, syntaxiques) au sein d'une structure dite « traits d'unification » (Joshi *et al.*, 1975) associés à chaque nœud de l'arbre élémentaire. En plus de ces traits d'unification, elle a défini un ensemble de « traits d'instanciation ». Ces derniers sont attribués seulement aux nœuds ancres et définissent les informations morphosyntaxiques nécessaires lors de la lexicalisation de la grammaire.
ArabTAG est partiellement lexicalisée puisqu'elle contient un ensemble d'arbres élémentaires lexicalisés réservés pour représenter les contextes possibles des mots outils jouant le rôle de « modificateurs » des phrases et/ou des syntagmes.
La première version d'ArabTAG est composée de différentes structures syntaxiques réparties entre

phrases et syntagmes. Elle contient 380 arbres élémentaires différents, répartis entre quatre grandes familles : phrases verbales, phrases nominales, syntagmes nominaux et syntagmes prépositionnels. Elle couvre quelques structures elliptiques, d'autres anaphoriques et aussi des structures renfermant des subordonnées. Elle prend en considération la variation de l'ordre des éléments au sein des phrases et aussi le phénomène d'agglutination.
Nous avons étudié cette première version de la grammaire et nous avons relevé certaines limites pouvant se résumer comme suit :

— une couverture minimale : les structures syntaxiques possibles ne sont pas toutes décrites. Elle ne représente pas, par exemple, les structures enrichies avec des compléments (circonstanciel de temps, de lieu, etc).
— la représentation des formes agglutinantes dans les structures syntaxiques n'est pas bien prise en compte. Ces formes peuvent jouer des rôles dans la phrase et doivent être mises en relief pour améliorer la couverture du modèle grammatical développé.
— ArabTAG V1.0 met l'accent sur les relations syntaxiques sans s'intéresser aux informations sémantiques, bien que la sémantique, tout comme la morphologie, possède une influence directe sur la syntaxe. En effet, l'interprétation syntaxique ne peut être complète que si l'on fait intervenir des informations sémantiques.
— ArabTAG V1.0 n'est pas organisée en des structures factorisées hiérarchiquement. Elle est composée d'un ensemble d'arbres élémentaires sans qu'ils soient reliés entre eux.Dans le but de faciliter la maintenance et l'extension de la grammaire, il est primordial de structurer la grammaire en faisant intervenir divers phénomènes tels que l'héritage des structures ou la hiérarchie des patrons d'arbres.

Par conséquent, nous proposons une nouvelle version ArabTAG V2.0 afin d'intégrer davantage les aspects syntaxiques et bénéficier des avantages représentatifs de TAG.

3 Description de l'arabe au moyen d'une méta-grammaire

La nouvelle version de la grammaire : ArabTAG V2.0 (Ben Khelil *et al.*, 2016) est réécrite en utilisant le formalisme XMG (Crabbé *et al.*, 2013). Ce dernier présente des caractéristiques particulièrement pertinentes pour la description des arbres élémentaires pour la langue arabe :

— il est très expressif, ce qui permet de définir des descriptions factorisées de la grammaire (dans notre cas, il a été utilisé pour traiter l'ordre des mots semi-libre).
— il est particulièrement adapté à la description des grammaires d'arbres et a été utilisé pour développer plusieurs grammaires TAG électroniques comme par exemple le français [5], l'anglais [6] et l'allemand [7] ;
— il est extensible et peut être configuré pour décrire différents niveaux de langage, comme la sémantique ou la morphologie.

A l'aide de ce formalisme, nous avons généré notre TAG semi-automatique pour l'arabe à partir d'une description réduite des règles de la grammaire. Cette description compacte de l'information grammaticale correspond à une méta-grammaire. Elle capture les généralisations linguistiques apparaissant parmi les arbres de la grammaire et permet ensuite de générer les arbres TAGs correspondants.
ArabTAG V2 est organisée en des structures factorisées hiérarchiquement grâce au mécanisme d'héritage de XMG. Ce dernier permet d'automatiser la combinaison des fragments d'arbres, que nous

5. https ://sourcesup.renater.fr/xmg/frenchmetagrammar/index.html
6. http ://homepages.inf.ed.ac.uk/s0896251/XMG-basedXTAG/titlepage.html
7. http ://www.sfs.uni-tuebingen.de/emmy/res.html

avons définis, en attribuant une couleur (B, W et R pour noir, blanc et rouge respectivement) aux nœuds. La couleur rajoute une contrainte de bonne formation des arbres engendrés : Un nœud coloré en noir peut être unifié avec 0, 1 ou plusieurs nœuds blancs et produit ainsi un nœud noir. Un nœud blanc doit être unifié avec un noir produisant un nœud noir. Finalement, un nœud rouge ne peut pas être fusionné avec un autre nœud.
Nous avons commencé par définir une classe EpineVerbe(C) [8] qui contribue à un fragment d'arbre pour l'épine verbale (classe MorphActive). Nous avons rajouté des points d'adjonction appropriés pour les adverbes [9]. Ces derniers peuvent être librement intercalés entre arguments.

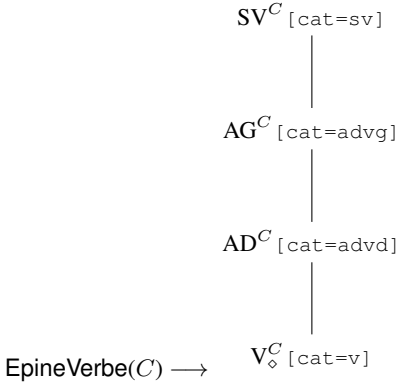

FIGURE 4 – Description de la classe EpineVerbe

La classe MorphActive instancie EpineVerbe. La valeur de la couleur passée en paramètre est noir :

MorphActive $\longrightarrow$ EpineVerbe(B)

La classe SujetCanon représente le sujet du verbe (cas nominatif) :

SujetCanon $\longrightarrow$ [AD] $\Leftarrow$ EpineArg() $\wedge$ SN$^{\mathrm{R}}_{\downarrow}$ [cat=sn,cas=nom] $\wedge$ AD $\triangleleft$ SN

Quant aux compléments d'objets directs et indirects ils sont définis comme suit : Le complément d'objet direct est exprimé par un syntagme nominal [cat=sn] dont le cas est accusatif [cas=acc]. Ce complément d'objet peut se placer avant le verbe. Dans ce cas, une anaphore (référant au complément d'objet) doit s'attacher à la fin du verbe. Cette contrainte est exprimée avec le trait [oclit=+] [10] :

ObjetCanonSN $\longrightarrow$ [AD, V] $\Leftarrow$ EpineArg() $\wedge$ SN$^{\mathrm{R}}_{\downarrow}$ [cat=sn, cas=acc] $\wedge$ AD $\triangleleft$ SN
$\wedge$ ((V [oclit=−] $\wedge$ V $\prec^{+}$ SN) $\vee$ (V [oclit=+] $\wedge$ SN $\prec^{+}$ V))

La classe suivante indique que le complément d'objet peut être un enclitique au verbe :

ObjetCanonClit $\longrightarrow$ [V] $\Leftarrow$ EpineVerbe(W) $\wedge$V [oclit=+]

Quant au complément d'objet indirect, il est exprimé par un syntagme prépositionnel :

ObjetIndCanon $\longrightarrow$ [AD] $\Leftarrow$ EpineArg() $\wedge$ SP$^{\mathrm{B}}$ [cat=sp] $\wedge$ AD
P$^{\mathrm{R}}_{\diamond}$ SN$^{\mathrm{R}}_{\downarrow}$ [cat=sn, cas=gen]
$\triangleleft$ SP

8. EpineVerbe (Figure 4) est paramétré par une couleur C
9. AG pour adverbe à gauche et AD pour adverbe à droite
10. Trait qui indique la présence d'un enclitique liée au nœud en question.

Au final, nous avons utilisé la transitivité du verbe comme un critère fondamental pour l'héritage et nous avons combiné [11] ces fragments afin d'obtenir les trois familles de verbes (intransitif, transitif, ditransitif). Chacune de ces classes capture les différentes réalisations syntaxiques possibles entre les différentes structures de la phrase. Elles sont organisées comme suit :
Intransitive $\rightarrow$ MorphActive $\wedge$ (SujetCanon $\vee$ Ellipse)
La classe Morphactive définit le fragment d'arbre élémentaire qui constitue l'épine verbale de la phrase. Suite à une opération de conjonction entre cette classe et les classes du sujet (SujetCanon ou Ellipse du sujet), nous obtenons la classe Intransitive.
Transitive $\rightarrow$ Intransitive $\wedge$ ObjetCanon[Objet1] [12]
La classe Transitive est obtenue en combinant la classe Intransitive est la classe ObjetCanon. Cette dernière représente une disjonction entre le complément d'objet direct, indirect et clitique.
DiTransitive $\rightarrow$ Transitive $\wedge$ ObjetCanon[Objet2] [13]
Finalement, nous obtenons la classe DiTransitive en combinant Transitive avec les compléments d'objets seconds.
L'utilisation de XMG nous a permis de gérer facilement le problème d'ordre semi-libre des mots. Pour ce faire, nous avons évité d'imposer une contrainte de précédence entre les arguments dont le changement d'ordre n'affecte pas la cohérence de la phrase.
Jusqu'à présent, nous avons généré 315 arbres à partir d'une description faite de 28 classes (soit 28 fragments d'arbres ou règles de combinaison). Cette version de la grammaire couvre les structures déjà couvertes par la première version d'ArabTAG à savoir les phrases verbales (forme active et passive), phrases nominales, syntagmes nominaux et syntagmes prépositionnels. La couverture de la grammaire a été également étendue en ajoutant des arbres élémentaires pour la représentation des compléments tels que les compléments circonstanciels de temps, compléments circonstanciels de lieu et les adverbes.
Nous avons mis en place un environnement de développement (script en langage Python) afin de vérifier la couverture grammaticale d'ArabTAG V2.0. En plus de notre grammaire, nous avons défini des lexiques syntaxiques et morphologiques en suivant l'architecture en 3 couches du projet (XTAG Research, 2001). En effet, le système XTAG se compose de trois sous-modules :

- Une base de schèmes qui sont classés en familles d'arbres élémentaires.
- Une base de lemmes où chaque lemme est associé une (ou plusieurs) famille(s) d'arbres et à un prédicat sémantique.
- Une base morphologique dans laquelle chaque forme fléchie est associée à un lemme et l'information morphosyntaxique appropriée.

Le but de ce test (Figure 5) est d'évaluer à la fois la sous-génération et la surgénération. A chaque nouveau phénomène syntaxique inclus dans ArabTAG V2.0, le corpus de test est enrichi manuellement avec des phrases grammaticales et non grammaticales (associées aux nombres d'analyses syntagmatiques attendues). L'analyseur TuLiPA (Parmentier *et al.*, 2008) est ensuite exécuté sur le corpus de test pour vérifier la qualité de la grammaire. Les résultats des analyses nous aident à corriger les erreurs potentielles dans la description de notre grammaire et nous garantit la cohérence des structures TAG lors de son extension.

11. XMG offre des opérations de combinaisons par disjonction et conjonction des fragments
12. ObjetCanon[Objet1] $\longrightarrow$ ObjetCanonSN[Objet1] $\vee$ ObjetCanonClit[Objet1] $\vee$ ObjetIndCanon[Objet1]
13. ObjetCanon[Objet2] $\longrightarrow$ ObjetCanonSN[Objet2] $\vee$ ObjetCanonClit[Objet2] $\vee$ ObjetIndCanon[Objet2]

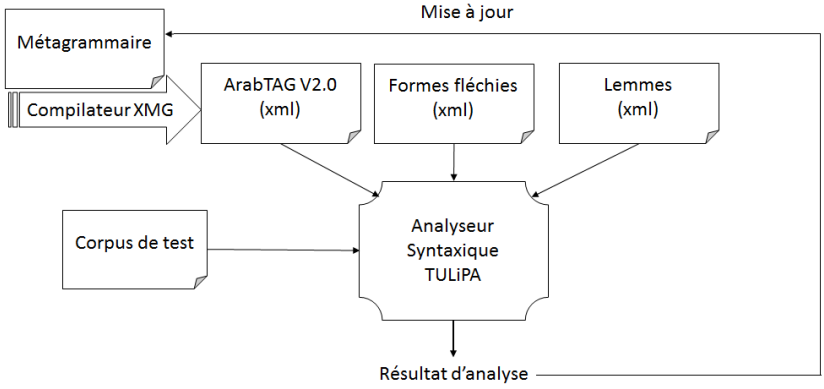

FIGURE 5 – Architecture du module de validation d'ArabTAG v2.0

4 Évaluer la qualité de la grammaire

L'étape d'évaluation est primordiale et doit prendre en compte un certain nombre de critères. Il est indispensable de s'interroger non seulement sur les ressources nécessaires à une telle évaluation mais aussi la procédure à adopter et quels critères d'évaluation utiliser.

4.1 Critères d'évaluation

Tout d'abord, et avant de s'engager dans cette étape, nous avons fait le point sur ce que notre grammaire doit satisfaire. En effet, la grammaire doit pouvoir :

- Distinguer les phrases valides telles qu'elles sont décrites dans l'arabe standard moderne (livres scolaires, romans arabes, etc.)
- Distinguer les phrases agrammaticales
- Analyser un ensemble assez grand de phrases qui couvrent les phénomènes syntaxiques importants de l'arabe.

Nous avons commencé par définir les critères de l'évaluation. Les métriques classiques utilisées sont la précision et le rappel. La précision est le nombre de phrases grammaticales bien analysées par rapport au nombre total de phrases analysées. Quant au rappel, il est défini par le nombre de phrases grammaticales bien analysées par rapport au nombre total de phrase grammaticales présentes dans le corpus de test. Ces deux métriques sont combinées dans une mesure nommée F-mesure (ou encore le f-score). La quantification de ces critères se fait donc sur la base d'un corpus annoté, typiquement un corpus arboré.

4.2 Protocole d'évaluation

Les outils dont nous disposons (présentés dans la section précédente) ne peuvent pas supporter un lexique de grande taille, ce qui ne nous permet pas de conclure sur la qualité de la grammaire. En effet, nous avons défini le lexique et le corpus de test manuellement. Le corpus de test est constitué de 120 phrases et syntagmes dont 32 agrammaticales. Nous voulons donc évaluer notre grammaire en utilisant un corpus arboré de grande taille. Le jeu de test sera constitué d'un ensemble de phrases et de syntagmes. L'étape suivante consistera à extraire automatiquement à partir du corpus les éléments nécessaires pour constituer un lexique compatible avec notre grammaire.

4.3 Ressources et Corpus d'évaluation

La difficulté majeure que nous avons rencontrée dans cette étape concerne les ressources nécessaires à l'évaluation. En effet, la disponibilité de ces ressources est contrastée selon les langues. Pour l'arabe standard, un ensemble de trois corpus arborés a été créé : Penn Arabic TreeBank [14](PATB) (Maamouri *et al.*, 2004),Prague Arabic Dependency Treebank [15] (PADT) (Hajič *et al.*, 2004) et Columbia Arabic Treebank [16] (CATiB) (Habash & Ryan, 2009). Ces corpus n'adoptent pas le même format d'annotation. PATB repose sur les représentations à base des structures de constituants [17] tandis que PADT et CATiB suivent, plutôt, un format à base des structures de dépendances [18].Les textes inclus dans ces corpus sont des textes journalistiques. Or nous voulons exploiter des textes littéraires renfermant des structures plus riches et plus représentatives de l'arabe et de ses phénomènes syntaxiques. En plus, tous ces corpus ne sont pas des ressources libres et leurs coûts sont élevés. Il nous est difficile de les utiliser dans notre évaluation.

4.4 Proposition pour ArabTAG V2.0

Face à la contrainte de disponibilité des ressources pour l'évaluation, nous avons décidé d'utiliser un ensemble de textes étiquetés. Ces textes, initialement bruts [19], ont subi un ensemble de traitements séquentiels [20] qui ont permis de les transformer en des phrases étiquetées par les arbres syntaxiques corrects qui leur correspondent (Ben Fraj *et al.*, 2009). Ce corpus a été construit et utilisé dans la procédure d'analyse syntaxique à base d'apprentissage pour la langue arabe (Ben Fraj, 2010).
Ces textes étiquetés seront le point de départ pour construite un corpus comportant des phénomènes précis de l'arabe. Nous pouvons citer l'un des corpus de phénomènes syntaxiques de l'anglais Phenomenal Corpus (Letcher & Baldwin, 2013) qui a été annoté manuellement. Chaque phrase de notre corpus sera associée à un phénomène syntaxique de l'arabe que nous voulons tester. L'étape suivante consistera à extraire automatiquement, à partir du corpus étiqueté, les éléments nécessaires à notre lexique, à savoir la liste des formes fléchies avec leurs traits ainsi que la liste des lemmes

14. https ://catalog.ldc.upenn.edu/LDC2003T06
15. https ://ufal.mff.cuni.cz/padt/PADT_1.0/docs/index.html
16. http ://www1.ccls.columbia.edu/ CATiB/Home.html
17. Les données sont étiquetées à trois niveaux : morphologique, syntaxique et grammatical.
18. Ces corpus utilisent un format de dépendances simple étiqueté par les relations fonctionnelles des différents composants syntaxiques.
19. Les textes sources ont été collectés du livre de la langue arabe de la 8ème année de base de l'enseignement tunisien (2007-2008).
20. une analyse morphosyntaxique, un étiquetage grammatical, une segmentation en phrases et finalement un étiquetage syntaxique. Certains de ces pré-traitements ont été suivis d'une vérification manuelle.

correspondants. Les traits seront automatiquement (ou semi-automatiquement) attribués lors de l'extraction grâce à un procédé de correspondance entre les jeux d'étiquettes du corpus et ceux utilisés dans notre grammaire. Par conséquent, ce lexique est spécifique à notre corpus. Néanmoins, nous envisageons son enrichissement en utilisant un dictionnaire arabe. Le corpus sera agrandi avec des nouveaux cas plus complexes ainsi que des phrases agrammaticales.

5 Conclusion

L'analyse syntaxique de la langue arabe est une tâche difficile à entreprendre. Cela explique la rareté des ressources numériques disponibles pour le traitement automatique de cette langue ; à savoir les grammaires et les corpus. Dans cet article, nous avons présenté une nouvelle approche visant à construire une grammaire d'arbres adjoints pour représenter la syntaxe de l'arabe. Cette grammaire est réécrite en utilisant un langage de description méta grammatical XMG. Ce langage extensible fournit une représentation compacte de l'information grammaticale et offre un mécanisme pour combiner les fragments élémentaires d'information. Il permet ainsi le partage des informations entre les structures grammaticales et peut être configuré pour décrire différents niveaux du langage.
La grammaire couvre les phrases verbales (forme active et passive), phrases nominales, syntagmes nominaux et syntagmes prépositionnels. Elle traite la variation des positions des éléments au sein des composants syntaxiques, les compléments supplémentaires et les formes agglutinées.
De plus, elle utilise les traits d'unification. Ces traits présentent des informations morphologiques, syntaxiques et syntaxico-sémantiques supplémentaires qui sont associées à un mot ou à un syntagme.
Actuellement, nous voulons évaluer notre grammaire. Cependant l'indisponibilité des ressources nécessaires pour une telle évaluation nous a contraints à utiliser un petit corpus de test qui sera agrandi avec d'autres exemples complexes de phrases.
Nous envisageons d'améliorer la couverture d'ArabTAG V2.0 en intégrant les dimensions morphologiques et sémantiques. Pour ce faire, nous étudions l'utilisation éventuelle de XMG2 (Petitjean, 2014) qui étend XMG en incluant un compilateur méta-grammatical. Ceci rend possible la description des nouvelles dimensions, qui sont respectivement la description de la morphologie en utilisant les champs topologiques et celle de la sémantique en utilisant les frames.

Remerciements

Nous remercions les relecteurs pour leurs commentaires pertinents qui ont permis d'améliorer la qualité de ce papier.

Références

ALKHULI M. (1979). *A contrastive transformational grammar : Arabic and English*. Leiden E. J. Brill.

BEN FRAJ F. (2010). *Un analyseur syntaxique pour les textes en langue arabe à base d'un apprentissage à partir des patrons d'arbres syntaxiques*. PhD thesis, ENSI La Manouba, Tunisia.

BEN FRAJ F. (2011). Construction d'une grammaire d'arbres adjoints pour la langue arabe. In *Actes de la 18e conférence sur le Traitement Automatique des Langues Naturelles*, Montpellier, France : Association pour le Traitement Automatique des Langues.

BEN FRAJ F., BEN OTHMANE ZRIBI C. & BEN AHMED M. (2009). A semi-automatic tag syntactic tagging tool for constructing an arabic treebank. In *Proceedings of the International Multiconference on Computer Science and Information Technology*, Mrągowo , Pologne.

BEN KHELIL C., DUCHIER D., PARMENTIER Y., ZRIBI C. & BEN FRAJ F. (2016). ArabTAG : from a Handcrafted to a Semi-automatically Generated TAG. In *TAG+12 : 12th International Workshop on Tree-Adjoining Grammars and Related Formalisms*, Düsseldorf, Germany.

CRABBÉ B., DUCHIER D., GARDENT C., LE ROUX J. & PARMENTIER Y. (2013). XMG : eXtensible MetaGrammar. *Computational Linguistics*, **39**(3), 591–629.

FEHRI A. (1981). *Complémentation et anaphore en arabe moderne : une approche lexicale fonctionnele*. Doctoral dissertation.

HABASH N. & RAMBOW O. (2004). Extracting a tree adjoining grammar from the penn arabic treebank. *Proceedings of Traitement Automatique du Langage Naturel (TALN-04)*, p. 277–284.

HABASH N. & RYAN M. R. (2009). Catib : The columbia arabic treebank. *Proceedings of the ACL-IJCNLP 2009 Conference Short Papers, 2009 ACL and AFNLP, Suntec, Singapore*.

HADDAR K., BOUKEDI S. & ZALILA I. (2010). Construction of an hpsg grammar for the arabic language and its specification in tdl. *International Journal on Information and Communication Technologies*, **3**(3).

HAJIČ J., SMRŽ O., ZEMÁNEK P., ŠNAIDAUF J. & BEŠKA E. (2004). Prague arabic dependency treebank : Development in data and tools. *In Proceedings of the NEMLAR International Conference on Arabic Language Resources and Tools,Cairo, Egypt*.

JOSHI A., LEVY L. & TAKAHASHI M. (1975). Tree adjunct grammars. *Journal of Computer and System Sciences*, **10**(1), 136 – 163.

KEBBE M. (2000). *Transformational Grammar of Modern Literary Arabic*. Routledge.

LETCHER N. & BALDWIN T. (2013). Constructing a phenomenal corpus : Towards detecting linguistic phenomena in precision grammars. In *Workshop on High-level Methodologies for Grammar Engineering @ ESSLLI 2013*, p. 367–376, Düsseldorf, Germany.

MAAMOURI M., BIES A., BUCKWALTER T. & MEKKI. W. (2004). The penn arabic treebank : Building a large-scale annotated arabic corpus. *NEMLAR International Conference on Arabic Language Resources and Tools*.

MUTAWA A., ALNAJEM S. & ALZHOURI F. (2008). An hpsg approach to arabic nominal sentences. *JASIST*, **59**(3).

PARMENTIER Y., KALLMEYER L., LICHTE T., MAIER W. & DELLERT J. (2008). TuLiPA : A Syntax-Semantics Parsing Environment for Mildly Context-Sensitive Formalisms. In *9th International Workshop on Tree-Adjoining Grammar and Related Formalisms (TAG+9)*, p. 121–128, Tübingen, Germany.

PETITJEAN S. (2014). *Génération Modulaire de Grammaires Formelles*. PhD thesis, Université d'Orléans, France.

WEIR D. J. (1988). *Characterizing Mildly Context-Sensitive Grammar Formalisms*. PhD thesis, Université de Pennsylvanie, États-Unis.

XTAG RESEARCH G. (2001). A lexicalized tree adjoining grammar for english. *Technical Report IRCS-01-03, IRCS, University of Pennsylvania*.