Extraction de grammaires TAG lexicalisées avec traits à partir d'un corpus arboré pour le coréen

Jungyeul Park

Université Paris 7 – Denis Diderot – UFRL – Laboratoire de linguistique formelle
jungyeul.park@linguist.jussieu.fr

Résumé

Nous présentons, ici, une implémentation d'un système qui n'extrait pas seulement une grammaire lexicalisée (LTAG), mais aussi une grammaire LTAG avec traits (FB-LTAG) à partir d'un corpus arboré. Nous montrons les expérimentations pratiques où nous extrayons les grammaires TAG à partir du *Sejong* Treebank pour le coréen. Avant tout, les 57 étiquettes syntaxiques et les analyses morphologiques dans le corpus SJTree nous permettent d'extraire les traits syntaxiques automatiquement. De plus, nous modifions le corpus pour l'extraction d'une grammaire lexicalisée et convertissons les grammaires lexicalisées en schémas d'arbre pour résoudre le problème de la couverture lexicale limitée des grammaires lexicalisées extraites.

Mots-clés : grammaire d'arbre d'adjoint lexicalisée, LTAG, LTAG avec traits, FB-LTAG, structure des traits, corpus arboré, extraction automatique d'une grammaire, coréen.

Abstract

We present the implementation of a system which extracts not only lexicalized grammars but also feature-based lexicalized grammars from *Sejong* Treebank for Korean. We report on some practical experiments, in which we extract TAG grammars. Above all, full-scale syntactic tags and well-formed morphological analysis in Sejong Treebank allow us to extract syntactic features. In addition, we modify the Treebank to extract lexicalized grammars and convert them into tree schemata to resolve limited lexical coverage problems related to extracted lexicalized grammars.

Keywords: lexicalized tree adjoining grammar, LTAG, feature-based LTAG, feature structure, treebank, automatic grammar extraction, Korean.

1. Introduction

La grammaire électronique est une interface entre la complexité et la diversité de la langue naturelle et la régularité et l'efficacité du traitement automatique de la langue (TAL). Elle est une des ressources les plus importantes pour le traitement automatique des langues naturelles (Abeillé et Blache 2000 ; Abeillé, 2002). Une grammaire à grande échelle peut incorporer beaucoup d'informations concernant le lexique, la syntaxe et la sémantique d'une langue humaine et un énorme effort humain est nécessaire pour la construire et la maintenir. Parce que le développement manuel de la grammaire est une tâche intensive qui prend beaucoup de temps, beaucoup d'efforts pour le développement semi-automatique et automatique de la grammaire ont été fournis pendant la décennie dernière (Candito, 1999 ; Xia, 2001 ; Chen, 2000 ; Nasr, 2004 ; Johansen, 2004 ; Habash et Rambow, 2004 ; Chiang, 2000 ; Xia *et al*., 2000 ; Neumann, 2003 ; Frank, 2001).

Puisque nous pouvons extraire une grammaire automatiquement sans beaucoup d'efforts, si le corpus arboré fiable est fourni, nous réalisons un système qui extrait une grammaire d'arbres adjoints lexicalisée et une grammaire d'arbres adjoints lexicalisée avec les traits, à partir du corpus arboré *Sejong* (SJTree) pour le coréen dans cet article. Le corpus SJTree contient 32 054 *eojeol*s (l'unité de la segmentation dans la phrase coréenne) c'est-à-dire, 2 526 phrases

avec 69 240 morphèmes. Le corpus SJTree emploie les 43 étiquettes de partie de discours et les 57 étiquettes syntaxiques.

Parmi les grammaires extraites automatiquement, on peut citer celle de Chen (2000) qui extrait une grammaire TAG lexicalisée à partir du corpus Penn Treebank et celles des autres travaux basés sur la procédure de Chen (2000) telles que Nasr (2004) et Johansen (2004) pour le français et Habash et Rambow (2004) pour l'arabe, celle de Chiang (2000) qui extrait une variation de TAG telle qu'une grammaire d'insertion d'arbres, celles de Xia *et al*. (2000) qui développe un système d'extraction des grammaires TAG lexicalisées pour l'anglais, le chinois et le coréen, celles de Neumann (1998 et 2003) qui extrait une grammaire à partir du corpus Penn Treebank pour l'anglais et à partir du corpus NEGRA pour l'allemand, et celle de Frank (2001) qui applique sa méthode d'extraction au corpus allemand NEGRA.

Bien qu'il y ait des travaux d'extraction d'une grammaire lexicalisée, l'extraction des traits n'est pas encore réalisée. Les 57 étiquettes syntaxiques et les analyses morphologiques dans le corpus SJTree nous permettent d'extraire les traits syntaxiques automatiquement et de développer une grammaire d'arbres adjoints lexicalisée avec traits. D'abord, nous expliquons notre procédure d'extraction et faisons un rapport sur le résultat expérimental. Enfin, nous discutons la conclusion.

2. Procédure d'extraction

Avant l'extraction automatique d'une grammaire TAG lexicalisée, nous transformons la structure des phrases annotées par parenthèses dans le corpus SJTree en structure d'arbres. Ensuite, nous prenons un algorithme « profondeur d'abord » pour le parcours d'arbre dans la structure des données et nous déterminons la tête et le type de l'opération (substitution ou adjonction) pour les nœuds des fils si le nœud donné est un nœud non-terminal.

2.1. Détermination de la tête

Pour la détermination de la tête, nous supposons que le nœud le plus à droite est une tête parmi les autres frères dans les langues à tête finale comme le coréen. Par exemple, dans la composition [NP NP], le deuxième NP est une tête alors que le premier NP est marqué comme une opération d'adjonction et extrait comme un arbre auxiliaire dans la grammaire G_1 qui utilise les *eojeol*s directement sans modification du corpus SJTree (voir la section 3 pour les détails de l'expérience d'extraction).

(1)

일본	외무성은	즉각	해명	성명을	발표했다
ilbon	*oimuseong.eun*	*jeukgak*	*haemyeng*	*seongmyeng.eul*	*balpyoha.eoss.da*
Japan	ministy_of_foreign_affairs.Nom	immediately	elucidation	declaration.Acc	annonce.Pass.Ter

The ministry of foreign affairs in Japan immediately announced their elucidation.

```
(S      (NP_SBJ     (NP ilbon/NNP)
                    (NP_SBJ oimuseong/NNG+eun/JX))
        (VP     (AP jeukgak/MAG)
                (VP     (NP_OBJ     (NP haemyeng/NNG)
                                    (NP_OBJ seongmyeng/NNG+eul/JKO))
                        (VP balpyo/NNG+ha/XSV+eoss/EP+da/EF+./SF))))
```

Figure 1. Phrase annotée par les parenthèses dans le corpus SJTree

De même, dans la composition [VP@VV VP@VX] où le premier VP a une ancre VV (verbe) et le deuxième VP a une ancre VX (verbe auxiliaire), le verbe principal (VV) est marqué comme une opération d'adjonction et extrait comme un arbre auxiliaire. Le verbe auxiliaire (VX) est une tête et il est extrait comme un arbre initial qui contient tous les arguments de la phrase. Ce phénomène est expliqué par la théorie « composition d'argument ». La détermination de la tête pour le verbe *balpyoha.eoss.da* ('avoir annoncé') en (1) est en gras dans la figure 1. La figure représente aussi, la phrase annotée par les parenthèses dans le corpus SJTree.

2.2. Détermination entre les opérations de substitution et d'adjonction

Différemment des autres corpora arborés comme le Penn Treebank pour l'anglais et le Paris 7 Treebank pour le français, les 57 étiquettes syntaxiques du corpus SJTree nous permettent de déterminer quels sont les nœuds marqués pour les opérations de substitution et d'adjonction. Parmi les 57 étiquettes syntaxiques, les syntagmes NP (le syntagme nominal), VNP (le syntagme de copule), VP (le syntagme verbal), et S (la phrase) qui finissent par le _CMP (l'attribut), _OBJ (l'objet), et _SBJ (le sujet) sont marqués comme une opération de substitution. Les nœuds étiquetés par les autres étiquettes, sauf la tête, sont marqués comme une opération d'adjonction. Dans cette distinction, il est possible que les syntagmes VNP et VP soient marqués comme une opération de substitution, ce qui veut dire que les syntagmes VNP et VP sont les arguments de la tête du syntagme car SJTree distribue les étiquettes VNP et VP au lieu de NP pour les formes de nominalisation de VNP et de VP.

Les nœuds fils marqués comme une opération de substitution sont remplacés par des nœuds terminaux de substitution (*e.g.* NP_SBJ↓) et appellent récursivement la procédure d'extraction avec le sous-arbre où le nœud racine est celui du nœud fils lui-même. Les nœuds fils marqués comme une opération d'adjonction sont supprimés de l'arbre principal et appellent récursivement la procédure d'extraction avec le sous-arbre où nous ajoutons le nœud racine avec celui du nœud parent du nœud donné et son frère (le nœud pied, *e.g.* VP*) avec celui du nœud donné. Comme il est défini dans le formalisme, la racine et le nœud pied du sous-arbre pour l'opération d'adjonction partagent la même étiquette.

2.3. Élagage

Les grammaires extraites expliquées au-dessous ne sont pas les grammaires TAG lexicalisées « correctes ». L'élagage signifie une réduction du tronc qui reste après avoir supprimé les nœuds d'adjonction. Après avoir supprimé les nœuds d'adjonction, il reste encore des nœuds dont nous n'avons pas besoin pour construire une grammaire TAG lexicalisée. La figure 2 montre cette opération de réduction du tronc pour le verbe *balpyoha.eoss.da* ('avoir annoncé') de la phrase (1).

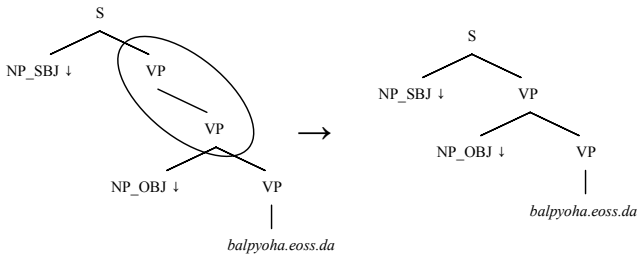


Figure 2. Procédure d'élagage

2.4. Extraction des traits

Les 57 étiquettes syntaxiques et les analyses morphologiques dans le corpus SJTree nous permettent d'extraire les traits syntaxiques automatiquement et de développer une grammaire d'arbres adjoints lexicalisée avec traits.

La grammaire FB-LTAG extraite emploie un ensemble d'étiquettes réduit parce qu'elle contient leur information syntaxique dans la structure des traits. Par exemple, une étiquette syntaxique NP_SBJ dans la grammaire LTAG est changée en NP et le trait syntaxique [<cas> = sujet] est ajouté. Par conséquent, nous utilisons 13 étiquettes syntaxiques pour la grammaire FB-LTAG. À partir des étiquettes syntaxiques qui finissent par _SBJ (le sujet), _OBJ (l'objet), et _CMP (l'attribut), nous pouvons extraire le trait <cas> qui décrit une structure des arguments dans la phrase. C'est-à-dire, le système extrait les traits <cas> à partir des arbres élémentaires d'une grammaire lexicalisée extraite pour les verbes et les adjectifs qui contiennent les étiquettes syntaxiques des arguments.

À côté du trait <cas>, nous pouvons aussi extraire les traits <mode> et <temps> à partir du corpus SJTree. Puisque les analyses morphologiques pour les terminaisons verbales et adjectivales dans le corpus SJTree sont simplement divisées en étiquettes morphologiques EP, EF et EC qui signifient, respectivement, une terminaison non-finale, une terminaison finale et une terminaison conjonctive, les traits <mode> et <temps> ne sont pas extraits directement à partir du corpus SJTree. Dans cet article, nous pré-analysons 7 terminaisons non-finales (EP) et 77 terminaisons finales (EF) utilisées dans le corpus SJTree pour extraire les traits <mode> et <temps> automatiquement. En général, l'étiquette morphologique EF porte la flexion du mode pour le trait <mode> et EP porte la flexion du temps pour le trait <temps>. Lorsque le système rencontre ces terminaisons non-finales ou finales pendant la procédure d'extraction, il réfère aux listes de EP et de EF pour extraire les traits <mode> et <temps>.

En général, les terminaisons conjonctives (EC) ne concernent pas l'extraction des traits <mode> et <temps>, nous extrayons seulement le trait <ec> qui contient la valeur de chaîne. Quelques terminaisons non-finales comme *si* sont extraites par le trait [<hor> = +] qui a un sens honorifique. Dans la grammaire FB-LTAG extraite, nous représentons la tête du lexique sous la forme infinitive et les traits contiennent la forme fléchie. Un arbre initial pour le verbe *balpyoha.eoss.da* ('avoir annoncé') de la figure 2 est changé comme la figure 3.

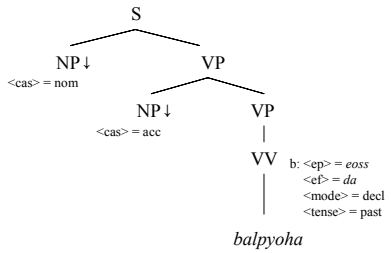


Figure 3. Un exemple d'une grammaire FB-LTAG pour balpyoha.eoss.da ('avoir annoncé)[1]

Le trait <det> est aussi automatiquement extractible dans le corpus SJTree et il est extrait à partir de l'étiquette syntaxique et de l'analyse morphologique, à la différence des autres traits

[1] En fait, la procédure d'extraction de la grammaire avec traits se déroule en 2 phases : la première est la conversion d'étiquettes syntaxiques du corpus en catégories plus traits, et la deuxième est l'ajout d'équations plus générales sur ces mêmes traits en se basant sur la notion d'épine dorsale. Par exemple, on joute les mêmes traits de VV aval sur les nœuds S aval, VP amont/aval et VV amont dans la figure 3 parce que les nœuds d'épine dorsal partagent un certain nombre de traits.

extraits[2]. Par exemple, tandis que le trait [<det> = -] est extrait à partir des noms dépendants qui ont toujours besoin d'un modifieur (extrait par l'analyse morphologique), le trait [<det> = +] est extrait à partir des modifieurs comme les syntagmes _MOD (extrait par l'étiquette syntaxique). À partir de l'étiquette syntaxique DP qui contient les MMs (déterminants ou démonstratifs), le trait [<det> = +] est aussi extrait[3].

3. Expérimentations d'extraction et résultat

3.1. Extraction d'une grammaire lexicalisée

Initialement, nous extrayons 18 146 arbres lexicalisés tels que 5 419 arbres initiaux et 12 727 arbres auxiliaires, directement à partir du corpus SJTree sans modification. Dans cette section, nous extrayons non seulement les arbres lexicalisés sans modification du corpus SJTree, mais aussi les grammaires avec modification du corpus en utilisant certaines contraintes pour améliorer les grammaires extraites et leur couverture lexicale.

- G_1 : Nous extrayons la grammaire en utilisant *eojeol*s directement sans modification du corpus SJTree
- G_2 : Nous séparons les symboles des *eojeol*s. Les symboles séparés sont extraits et divisés en α et β arbres basés sur leur type.
- G_3 : Dans le cas de la composition d'*eojeol* [NOM + POSTPOSITION], nous séparons les postpositions d'*eojeol*. Les postpositions séparées sont extraites comme des arbres initiaux. Les postpositions complexes qui contiennent deux ou plus postpositions sont aussi extraites comme un arbre initial.
- G_4 : Nous convertissons les arbres auxiliaires pour le syntagme nominal en arbres initiaux. Nous enlevons aussi les étiquettes syntaxiques des arbres initiaux dans le syntagme nominal (*e.g.* NP_SBJ → NP).

Dans les expérimentations d'extraction d'une grammaire lexicalisée ci-dessus, nous supposons que les grammaires supérieures résultent les grammaires inférieures. Par exemple, la séparation des postpositions dans la grammaire G_3 résulte de la séparation des symboles dans la grammaire G_2.

[2] Le nom de trait <det> ne signifie pas strictement un sens de déterminant. Il a un sens très général qui contient déterminant, modifieur, etc.

[3] Le coréen n'a pas besoin d'un trait <personne> comme en anglais et des traits <genre> ou <nombre> comme en français. Han *et al*. (2000) propose plusieurs traits de la grammaire FB-LTAG pour le coréen qui nous n'utilisons pas dans cet article comme <adv-pp>, <top> et <aux-pp> pour les noms et <clause-type> pour les prédicats. Tandis que les postpositions sont séparées à partir de l'*eojeol* pendant la procédure d'extraction d'une grammaire lexicalisée, Han *et al*. considère la combinaison des noms et postpositions comme un *eojeol* (voir la section 3). La séparation des postpositions à partir de l'*eojeol* est plus efficace pour un développement de la grammaire. Dans Han *et al*. (2000), le trait <adv-pp> contient simplement la valeur de chaîne des postpositions adverbiales. Le trait <aux-pp> ajoute un sens sémantique des postpositions auxiliaires comme *seulement*, *aussi*, etc. que nous ne pouvons pas extraire automatiquement à partir du corpus SJTree ou des autres corpus arborée pour le coréen car le corpus annoté syntaxiquement ne contient pas telle information sémantique. Le trait <top> marque la présence ou l'absence de marqueur topique en coréen comme *neun*, cependant le marqueur topique dans le corpus SJTree est annoté comme le sujet, c'est-à-dire que le seul trait [<cas> = sujet] est extrait à partir du marqueur topique. Le trait <clause-type> indique le type de phrases qui a des valeurs comme main ('phrase principale'), coord ('phrase coordinative'), subordi ('phrase subordonnée'), adnom ('phrase adnominale'), nominal ('phrase nominale') et aux-connect ('phrase auxiliaire'). Puisque la distinction du type de phrases est très vague, sauf la phrase principale, nous n'adoptons pas ce trait. Au lieu de cela, le trait <ef> est extrait si la phrase est principale et le trait <ec> est extrait pour les autres types de la phrase.

Nous additionnons 436 arbres initiaux et 32 arbres auxiliaires pour les symboles séparés dans la grammaire G_2 et 299 arbres auxiliaires pour les postpositions dans la grammaire G_3. Pour les arbres extraits des symboles et des postpositions, nous ajoutons les étiquettes syntaxiques SYM et POSTP comme un nœud non-terminal intermédiaire que le corpus SJTree n'utilise pas. Voir la figure 4 pour les exemples d'arbre extrait des symboles et des postpositions.

La raison principale pour laquelle nous séparons les symboles et les postpositions est bien indiquée dans le tableau 1.Toutes les fréquences moyennes (`sum(freq)/count(ltree)` où `sum(freq)` est un nombre total de fréquence de la grammaire extraite et `count(ltree)` est un nombre d'arbres lexicalisés des grammaires extraites ne dépassent pas 4. Cependant, les fréquences moyennes des symboles et des postpositions sont, respectivement, 13,80 et 31,59. Alors que les grammaires extraites ne garantissent pas toutes les possibilités de la composition d'*eojeol* tels que [syntagme + symbole] et [syntagme + postposition], nous décidons la séparation des symboles et des postpositions. Cette séparation implique la modification du corpus SJTree[4].

La figure 4 montre les grammaires extraites G_4 à partir de la phrase en (1). Théoriquement, l'ordre d'extraction suit l'ordre des mots dans la phrase.

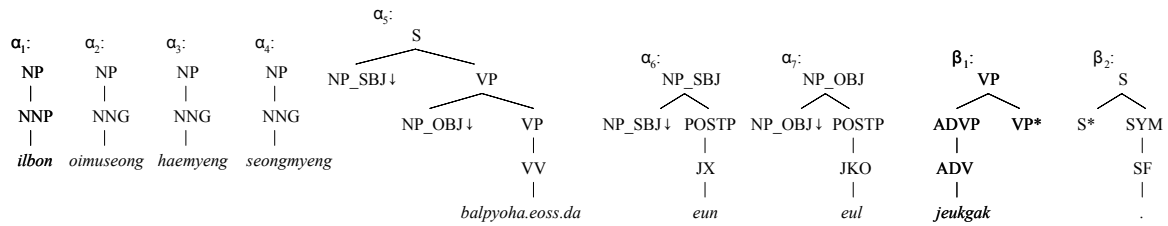


Figure 4. Grammaire lexicalisée G_4 pour la phrase en (1)

3.2. Extraction d'une grammaire lexicalisée avec traits

Nous extrayons une grammaire FB-LTAG qui emploie un ensemble d'étiquettes réduit parce qu'elle contient leur information syntaxique dans la structure des traits. La grammaire extraite G_5 enlève les étiquettes syntaxiques, utilise finalement un ensemble d'étiquettes réduit, ajoute la structure des traits, et introduit une forme infinitive pour son ancre lexicale.

- G_5 : Nous employons un ensemble d'étiquettes réduit et son ancre lexicale sous la forme infinitive avec une structure des traits.

La figure 5 montre les grammaires extraites G_5 à partir de la phrase en (1).

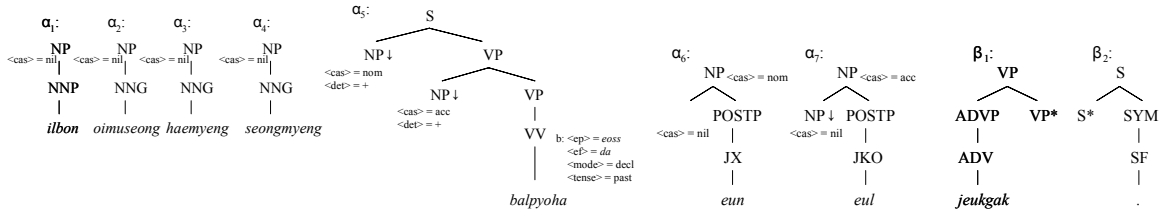


Figure 5. Grammaire lexicalisée avec traits G_5 pour la phrase en (1)

[4] La complexité du parseur par rapport à la taille de grammaire est indiquée dans Nasr (2004). Dans sa thèse d'habilitation, il a montré que la complexité est diminuée quand la taille de la grammaire est augmentée. Dans cet article, nous ne considérons pas la complexité du parseur, nous considérons seulement la couverture lexicale des grammaires extraites.

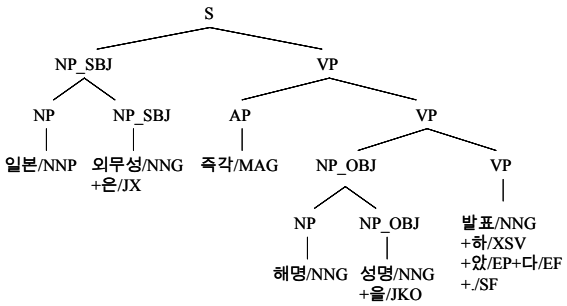
Le tableau 1 montre le résultat d'expérimentation d'extraction des grammaires extraites de G_1 à G_5.

	nombre de `ltrees`	fréquence s moyennes	60 % apprentissage (1511 phrases) et 40 % test (1015 phrases) : fréquence > 0	60 % apprentissage (1511 phrases) et 40 % test (1015 phrases) : fréquence > 1
G_1	18 146	1,38	0,731	0,806
G_2	18 015	1,70	0,724	0,802
G_3	17 485	2,28	0,742	0,820
G_4	13 373	2,98	0,737	0,808
G_5	12 239	3,26	0,811	0,870

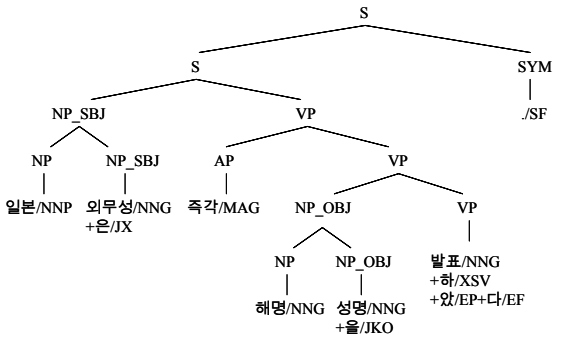
Tableau 1. Résultat d'expérimentation d'extraction

Et aussi, nous faisons une expérience de la couverture lexicale en appliquant les entrées lexicales des grammaires extraites à un corpus de 770 000 *eojeol*s qui est analysé morphologiquement. Après modification du SJTree, la couverture lexicale de G_3 est augmentée à 17,8 % par rapport à celle de G_1. Les grammaires G_4 et G_5 ont la même couverture lexicale que G_3 car elles partagent les mêmes entrées lexicales de G_3.

La figure 6 montre les analyses différentes avec les grammaires extraites différentes pour la phrase en (1). L'analyse avec la grammaire G_1 est montrée dans la figure 6a qui est tout à fait la même analyse du corpus SJTree, celle avec la grammaire G_2 dans la figure 6b montre une séparation du symbole et celle avec la grammaire G_3 dans la figure 6c montre une séparation des postpositions. Cependant, les grammaires G_4 ne peuvent êtres analysées comme dans la figure 6d seulement si nous adoptons les propositions de Park (2004) qui propose une structure plate intérieure pour les syntagmes nominaux et un *shallow* parseur avant l'analyse syntaxique. Nous avons aussi besoin de certaines contraintes comme « Tous les nœuds pour l'opération de substitution peuvent être substitués par n'importe quel syntagme nominal. » [5].

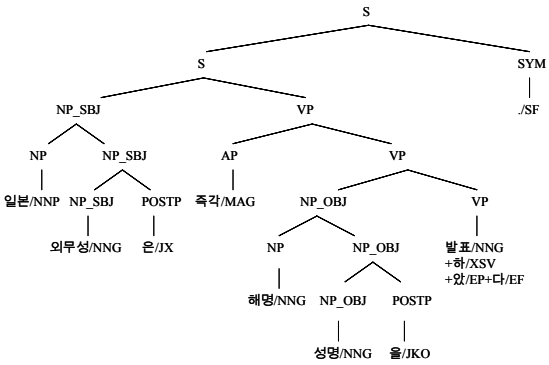


(a) Analyse avec G_1

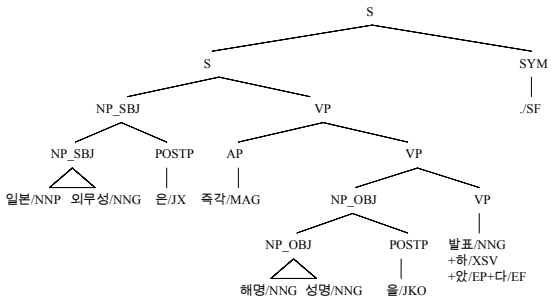


(b) Analyse avec G_2

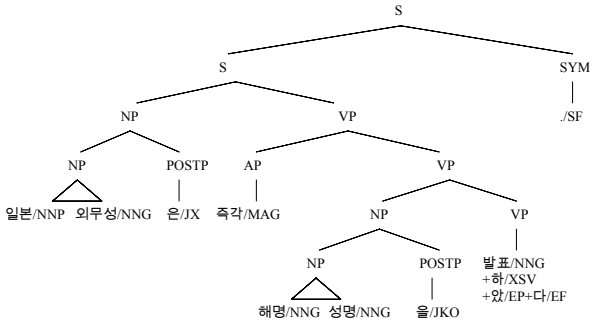
[5] Les analyses dans cet article ne sont pas les résultats par un parseur, mais proposées avec une supposition que le parseur utilise les grammaires extraites. Le parseur de Park (2004) emploie seulement la grammaire G_4 pour le moment.



(c) Analyse avec G_3



(d) Analyse avec G_4



(c) Analyse avec G_5

Figure 6. Analyses différentes selon la grammaire extraite

3.3. Extraction des schémas d'arbres élémentaires

Un des problèmes de l'extraction automatique d'une grammaire est le fait d'avoir une couverture lexicale limitée. Pour résoudre ce problème, nous élargissons les grammaires extraites en utilisant des gabarits appelés « schéma d'arbre » (Candito, 1999). L'ancre lexicale dans les grammaires extraites est enlevée et remplacée par un marqueur d'ancre pour former une grammaire de gabarit. Par exemple, un marqueur d'ancre @NNG où l'ancre lexicale des grammaires extraites est un nom commun qui remplace l'ancre lexicale. Le résultat d'expérimentation du converti en schéma d'arbre est montré dans le tableau 2.

	nombre de schémas d'arbre	fréquences moyennes
$G_1 \rightarrow T_1$	1 158	21,55
$G_2 \rightarrow T_2$	1 208	25,41
$G_3 \rightarrow T_3$	1 247	32,00
$G_4 \rightarrow T_4$	949	42,05
$G_5 \rightarrow T_5$	338	118,07

Tableau 2. Résultat d'expérimentation de la conversion en schéma d'arbre

3.4. Comparaison avec les familles d'arbres de Han *et al*. (2000)

Nous n'avons pas de grammaires lexicalisées faites manuellement pour le coréen dans le domaine public. Parmi 468 de T_5, nous comparons nos 234 schémas d'arbre pour les prédicats qui correspondent aux 14 sous-catégorisations avec les 11 sous-catégorisations de FB-LTAG proposée dans Han *et al*. (2000) pour évaluer la couverture de la grammaire. Nos schémas

d'arbres extraits automatiquement couvrent 72,7 % de la grammaire écrite à la main de Han *et al*. (2000)[6].

4. Conclusion

Dans cet article, nous présentons un système pour l'extraction qui produit une grammaire lexicalisée et aussi avec traits à partir d'un corpus arboré. Pour résoudre le problème de la couverture lexicale limitée des grammaires lexicalisées extraites, nous utilisons plusieurs options d'extraction, par exemple, la séparation des symboles et des postpositions, et puis la conversion de ces grammaires en schémas d'arbre qui ne contiennent pas d'ancre lexicale. Les grammaires extraites et les schémas d'arbre peuvent être utilisés dans les parseurs pour analyser les phrases coréennes et l'information de fréquence peut être utilisée pour enlever les ambiguïtés parmi les analyses syntaxiques possibles des parseurs.

Références

ABEILLÉ A. (2002). *Une grammaire électronique du français*. CNRS Éditions, Paris.

ABEILLÉ A., BLACHE Ph. (2000). « Grammaires et analyseurs syntaxiques ». In J.-M. Pierrel (éd.) *Ingénierie des Langues*. Hermès, Paris.

CANDITO M.-H. (1999). *Organisation modulaire et paramétrable de grammaires électroniques lexicalisées.* Thèse de doctorat, Université Paris 7.

CHEN J. (2001). *Towards Efficient Statistical Parsing Using Lexicalized Grammatical Information*. Thèse de doctorat, University of Delaware.

CHIANG D. (2000). « Statistical Parsing with an Automatically-Extracted Tree Adjoining Grammar ». In *Data Oriented Parsing*. CSLI Publication : 299-316.

FRANK A. (2001). « Treebank Conversion. Converting the NEGRA treebank to an LTAG grammar ». In *Proceedings of the Workshop on Multi-layer Corpus-based Analysis*. Iasi.

HABASH N., RAMBOW O. (2004). « Extracting a Tree Adjoining Grammar from the Penn Arabic Treebank ». In *Actes de TALN 2004.* Fès.

HAN C., YOON J., KIM N., PALMER M. (2000). *A Feature-Based Lexicalized Tree Adjoining Grammar for Korean*. IRCS Technical Report 00-04. University of Pennsylvania.

JOHANSEN A.D. (2004). *Extraction des grammaires LTAG à partir d'un corpus étiqueté syntaxiquement.* Mémoire de DEA, Université Paris 7.

NASR A. (2004). *Analyse syntaxique probaliliste pour grammaires de dépendances extraites automatiquement.* Habilitation à diriger des recherches, Université Paris 7.

NEUMANN G. (2003). « A Uniform Method for Automatically Extracting Stochastic Lexicalized Tree Grammar from Treebank and HPSG ». In A. ABEILLÉ (éd), *Treebanks*. Kluwer, Dordrecht.

PARK J. (2004). « Partially Lexicalized TAG Parser ». In *The 15th Meeting of Computational Linguistics in the Netherlands*. University of Leiden.

XIA F. (2001). *Automatic Grammar Generation from Two Different Perspectives*. Thèse de doctorat, University of Pennsylvania.

XIA F., PALMER M., JOSHI A. (2000). « A Uniform Method of Grammar Extraction and Its Application ». In *The Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora (EMNLP/VLC-2000)*. Hong Kong.

[6] C'est-à-dire, nous n'avons pas 3 sous-catégorisations de Han *et al*. (2000). Par ailleurs, Han *et al.* (2000) n'a pas 6 sous-catégorisations qui sont chez nous.