

Dissertation for the Degree of Doctor

Statistical Korean Dependency Parsing Model
based on the Surface Contextual Information

by

Hoojung Chung

Department of Computer Science and Engineering

Graduate School

Korea University

January 2004

林 海 彰 教 授 指 導
博 士 學 位 論 文

표층 문맥 정보에 기반한 통계적 한국어 의존분석 모델

Statistical Korean Dependency Parsing Model
based on the Surface Contextual Information

이 論文을 理學 博士學位 論文으로 提出함

2004 年 1 月

高麗大學校大學院

컴퓨터學科

丁厚中

丁厚中の 理學 博士學位 論文 審査를 完了함.

2004年 1月

委員長 임 해 창 (印)

委員 이 성 환 (印)

委員 육 동 석 (印)

委員 남 기 춘 (印)

委員 김 현 철 (印)

Abstract

Natural language parsing is a key problem to many tasks that require natural language processing. Many language processing tasks use the information on predicate-argument relation or modifier-modifyee relation and the parsing makes the extraction of the information possible by identifying relations between words, or phrases in sentences. However, it is difficult to parse a sentence correctly, because of the ambiguity inherent in the natural language. During the last decade, the statistical approach becomes the major trends in natural language parsing, or syntactic disambiguation. The two most important things in the statistical natural language parsing is selecting appropriate features that help syntactic disambiguation and designing a statistical model using them.

This dissertation argues that the influence of surface contextual information, such as modification distance and local context, in solving syntactic ambiguity of the Korean language, and proposes the parsing model that considers a modification distance in a certain local context in addition to the preference for lexical bigram dependency. All of these preferences are expressed by probabilities conditioned on local context. The parsing model is based on the dependency theory, which is widely known as an adequate formalism to reflect the syntactic characteristic of the Korean language, or other variable word-order languages. The statistical dependency parsing model consists of two probabilities, which are the lexical dependency probability and the modification distance probability; the lexical dependency probability reflects selectional preference and the preference on each dependency rules. The modification distance probability reflects the preferred length of a dependency relation from a certain modifier based on the context of the modifier.

We believe the parameterization of the parsing model for a language should be done with the deliberation of the characteristics of the language. The probability on modification distance is designed to consider the property of variable-word-order language, which includes Korean, and this is a new way to reflect the distance between two depending words. Evaluation on the KAIST Treebank text shows that the proposed model recovered dependency relations with 86.75% F_1 -score. The consideration of the modification distance and local context helps selecting correct modifyee of modifier even in variable-word-order language, and the proposed way to deal with the modification distance in the parsing model outperforms other methods dealing with the distance in the statistical model.

Contents

1	Introduction	1
1.1	Practical Motivation for Parsing	3
1.2	Statement of Thesis	5
1.3	Overview	6
2	Preliminaries	8
2.1	Dependency Grammar	8
2.2	A Brief Introduction to Korean Grammar	10
2.3	Parsing Korean with Dependency Grammar	12
2.3.1	Word Modeling	12
2.3.2	Dependency Rule	14
2.3.3	Constraints based on the Structural Properties of Korean	14
3	Previous Work	17
3.1	A Brief History of Parsing for Korean	17
3.2	Related Work	19
4	Parameterization for Dependency Parsing	25
4.1	Selection of $Event_i$	26
4.2	Selection of $Prediction_i$	26
4.3	Selection of $Context_i$	26
4.3.1	Part-of-Speech Tags of Depending Words	27

4.3.2	Lexical Bigram Dependency	28
4.3.3	Surrounding Words Information	29
4.4	Splitting the <i>Prediction_i</i>	31
4.4.1	Context for Modification Distance Prediction	31
4.4.2	Context for Word Dependency Prediction	35
4.5	Comparison with Other Models	36
5	Statistical Dependency Parsing Model for Korean	39
5.1	The Statistical Parsing Model	39
5.1.1	Lexical Dependency Probability	39
5.1.2	Modification Distance Probability	42
5.1.3	The Parsing Model	44
5.2	The Statistical Parser & Parameter Estimation	47
5.2.1	The Parser	47
5.2.2	Parameter Estimation	48
5.3	Experiments	52
5.3.1	Experimental Setup	52
5.3.2	Experiment 1 : Context-Window Size for Modification Distance Decision	54
5.3.3	Experiment 2 : Parsing Performance	57
5.3.4	Experiment 3 : Comparison with Other Parsers	59
5.4	Result Analysis	61
6	Discussion	68
6.1	Unlexicalized Parsing	68
6.1.1	Motivation	68
6.1.2	Investigation on Lexical Bigram Distribution	69
6.1.3	Unlexicalized Parser	72
6.2	Parsing with Splitted Part-of-speech tags	73
6.2.1	Word Clustering	73
6.2.2	Parsing With Splitted Tag	77

7 Conclusion	80
7.1 Contributions	81
7.2 Future works	81
A Functional Morpheme Abbreviation List	83
B Part-of-speech Tag Set	85
C Simple AUXP Chunker	88

List of Figures

1.1	A parsed tree	4
2.1	A dependency tree of the sentence <i>a man sleeps</i>	9
2.2	Another way to express the dependency structure	9
2.3	Dependency structure for the sentence <i>John Smith, the president of IBM announced his resignation yesterday</i>	10
2.4	Dependency trees for Korean sentences which have identical meaning, <i>Eugene watched a show yesterday</i>	12
2.5	Dependency relation between two Korean words	14
2.6	All dependents precede their heads	15
2.7	Wrongly analyzed sentence with crossing dependencies.	16
4.1	The dependency tree with part-of-speech tags for Example (4.5)	27
4.2	A dependency relation between <i>오래된</i> (<i>oraedoe-n</i> ; old) and <i>학교</i> (<i>hakgyo</i> ; school).	29
4.3	A wrong and a correct head for the word <i>오래된</i> (<i>oraedoe-n</i> ; old)	30
4.4	Two alternative modifyee candidates of <i>나의</i> (<i>na-ui</i> ; my)	31
4.5	The word ends with <i>의</i> (<i>-ui</i> ; -GEN) doesn't modify the next word because of the inserted modifier <i>아픈</i> (<i>apeun</i> ; sick)	33
4.6	Two alternative head candidates of <i>완전히</i> (<i>wanjeonhi</i> ; completely)	34
4.7	Dependency relations from <i>그리고</i> (<i>geurigo</i> ; and)	36
5.1	A dependency tree, which is part-of-speech tagged	46

5.2	Interpolations for estimating the lexical dependency probability	50
5.3	Procedure of back-off smoothing for modification distance.	51
5.4	Examples of AUXP chunking	53
5.5	Effect of different right contextual size	57
5.6	Effect of different K makes	58
5.7	Arc-based accuracy vs. length of dependency relations figures for the proposed model, Model3 and Model4 in the test data	65
5.8	Arc-based accuracy vs. length of dependency relations figures for the proposed model without outer context, Model3 and Model4 in the test data . . .	66
5.9	Effect of outer context in estimating lexical dependency	67
6.1	A distribution of word pairs	70
6.2	Distribution of the length of dependency relation from 국가 (<i>gukga</i> ; nation) and 사회 (<i>sahoe</i> ; society)	75
6.3	Distribution of the length of dependency relation from 경우 (<i>gyeongu</i> ; case) and 결과 (<i>gyeolgwa</i> ; result)	76
C.1	AUXP chunking rule set	90

List of Tables

1.1	Information extracted from the document about 김대중 (<i>Kim, Dae-Jung</i>) . . .	5
2.1	A part of the Korean dependency grammar	13
5.1	Lexical dependency probabilities between two words	41
5.2	Lexical dependency probabilities between two words showing the probabilities reflect the likelihood of grammar rules usage	42
5.3	Experimental result (in F_1 -score) for the modification distance classifier, with various m (left context size) , n (right context size) , and k (class size) values	55
5.4	Parsing Results on the training and testing data	59
5.5	Parsing Result for various parsers ($K = 4$ for Proposed model)	61
5.6	The result of 10-fold cross validation to the KAIST Treebank for Model4 and Proposed model	62
5.7	Effect of additional information	63
6.1	List of word ternaries that appeared only once in 27,694 sentences	71
6.2	Comparison of the parsing performance between the lexicalized and unlexi- calized parser	73
6.3	Comparison of the parsing performance between the lexicalized parser (Model4) and our unlexicalized parser	73
6.4	Result of part-of-speech tag splitting	78
6.5	Comparison of the parsing performance between the lexicalized and unlexi- calized parser	79

6.6	Parsing with combination of unlexicalized, splitted tagged, lexicalized information on the testing data	79
-----	---	----

Chapter 1

Introduction

Natural language parsing is a key problem to many natural language processing tasks such as machine translation, information extraction, and question answering. Higher accurate parsing helps such tasks to achieve better performance.

The parsing is a problem that maps any input sentence to an appropriate syntactic tree structure. It is a hard problem. Many decades have passed since the parsing research started, but the problem has not been satisfactorily solved yet.

Why is the natural language parsing so difficult? Simply, the answer is *ambiguity*. Natural language sentences have many structural ambiguities. Even human beings have difficulties in resolving ambiguities, while they are reading a sentence such as

He saw the man with the telescope.

, because the meaning of a natural language sentence varies depending on its context. Depending the given context of the above sentence, people can recognize where the prepositional phrase *with the telescope* modifies; *man* or *saw*.

A machine may face much more difficulties in resolving the syntactic ambiguity of natural language sentences, because it has little knowledge about the syntax of natural language. So the linguistic knowledge is encoded in the form of the grammar, lexicon, and ontology, and the machine performs disambiguation by using the selectional restrictions defined in them (Allen 1995). The selectional restrictions are specifications of the legal combinations of

words that can co-occur and be applied to eliminate wrong analysis constructed by a parser. However, selectional restrictions may have several problems, when it is used for parsing general-domain and wide-coverage tasks (Collins 1999). First of all, as the vocabulary size gets larger, the required linguistic information becomes larger. Thus, the incredible amount of manual labor is expected to build the whole necessary restriction set. Another problem is that the all-or-nothing nature of selectional restriction makes it impossible to encode semantic preferences. Further, in linguistics, the selectional restrictions have been known to have theoretical weakness.

Diverse statistical techniques have been suggested for solving these problems. Especially in 1990s, these approaches have become very popular, since a large syntactically annotated corpus, or Treebank, became available. Many probabilistic parsing models have been contrived to build robust and wide-coverage parsing systems. They don't require any hand-crafted grammar but the Treebank. They are trained and learn linguistic preferences from the annotated corpus, completely automatically. Some of these approaches were found to be successful and applied to improve performance of many natural language applications such as question answering (Chen, Diekema, Taffet, McCracken, Ozgencil, Yilmazel, and Liddy 2001; Xu, Licuanan, May, Miller, and Weischedel 2002), machine translation (Fox 2002; Yamada and Knight 2001), information extraction (Chelba and Mahajan 2001; Miller, Fox, Ramshaw, and Weischedel 2000) and so on.

Research on the statistical parsing model for Korean has also begun in 1990s. Kim (1994) illustrates a statistical language modeling for dependency parsing Korean. Kim and Seo (1997) proposed another statistical parser for Korean, which is a modified version of a statistical parser for English (Collins 1996) to reflect characteristics of the Korean language. Since there wasn't any available large-sized Treebank for Korean, they could train only from small-sized annotated corpus. As a Korean Treebank consists of 32,000 sentences (Choi 2001) had become available few years later, the more accurate statistical parsers using the corpus were suggested (Lee 1997a; Seo, Nam, and Choi 1999).

This dissertation proposes a new statistical parsing model for Korean. This thesis considers an important question regarding the statistical parsing model for Korean: *What linguistic feature is to be considered in disambiguating a syntactic structure of Korean effectively? And*

how can it be included in a statistical parsing model? We focus here on the *length of a modification relation* and *local contextual information* among many linguistic features. Despite of its importance and usefulness in syntactic disambiguation in the parsing of Korean, this feature has not dealt with seriously yet. Our goals are:

1. Verifying the usefulness of a distance measure in disambiguating syntactic structure of Korean, which is a variable word order language. Parameters related to the word order, such as the distance measure, are believed not to be suitable for parsing the variable word order languages by some researchers. We also inspect the effect of using other surface contextual information for syntactic disambiguation of Korean.
2. Designing a parsing model that formally considers the surface contextual information. The parsing model should reflect the property of the Korean language.
3. Advancing the state of the art of the Korean parsing by reporting improved parsing accuracy over previous results on an identical training and testing data.

1.1 Practical Motivation for Parsing

The parsing is a process that seeks an appropriate syntactic structure of a sentence. The tree in Figure 1.1 is a result of parsing the sentence *김대중은 1988년 국회의원에 당선되었고, 1991년 통합야당인 민주당을 창당하였다* (*Kim, Dae-Jung was elected a congressman in 1988 and founded Democratic Party, which was the merger of the opposition parties, in 1991*)¹. Using the information in the syntactic tree, text-processing application such as question answering, information extraction, and machine translation system can produce more accurate result. Two examples are introduced to show the usefulness of syntactic parsing in text processing applications.

The question answering system is our first example. The question answering system is an application that finds the exact answer of a given question from a user. The system usually

¹TOP, SUBCONJ, and ACC in the figure indicate the types of functional morphemes in Korean. The functional morpheme of the Korean language is explained in the next chapter. The full list of the abbreviations of the functional morpheme is on Appendix.

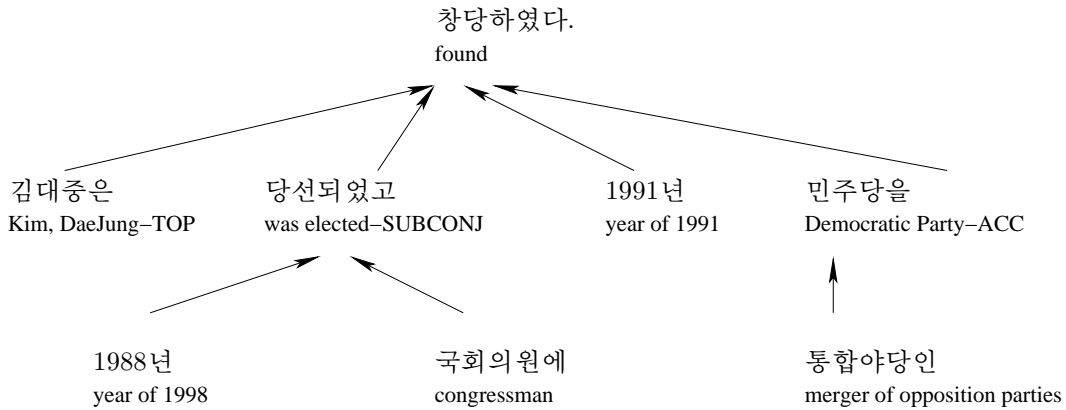


Figure 1.1: The parsed tree of the sentence 김대중은 1988년 국회의원에 당선되었고, 1991년 통합야당인 민주당을 창당하였다. (*Kim, Dae-Jung was elected a congressman in 1988 and founded Democratic Party, which was the merger of the opposition parties, in 1991*)

collects knowledge from documents in natural language text. If a question from a user is 김대중은 언제 민주당을 창당하였나? (*When did Kim, Dae-Jung found Democratic Party?*), the system has to find the year describing the event 창당하다 (*found*) in the sentence in Figure 1.1. There are two numbers in the sentence, which can be answer candidates of the question: 1988 and 1991. With the parse tree in Figure 1.1, the system can know that the year 1991 is related to the event 창당하다 (*found*) and answer correctly.

Information extraction is another example. Assume that we want to extract events from natural language documents automatically. An event consists of an event verb and entities related to it. Without parsed information, the information extraction system cannot find which argument associates with which event verb. Using the parse tree in Figure 1.1, the information extraction system can mine the information on the event verbs 당선되다 (*be elected*) and 창당하다 (*found*), as Table 1.1 shows.

The parsed result is important in other applications as well, if the applications require knowledge on language. Machine translation, high precision information retrieval, and speech recognition fall under this category (Collins 1999).

year	argument	event verb
1988	국회의원 (congressman)	당선되다 (be elected)
1991	민주당 (Democratic Party)	창당하다 (found)

Table 1.1: Information extracted from the document about 김대중 (*Kim, Dae-Jung*)

1.2 Statement of Thesis

This thesis deals with a statistical methodology to model the syntactic parsing of Korean. A statistical parsing model is built. Statistical parsing models assign a probability $P(t|S)$ to each tree t for the sentence S . Since Korean allows relatively free word order, the parsing model is designed to assign probabilities to dependency trees, not phrase structure trees. Dependency grammar, which defines formation of the dependency tree, has been widely used for generating the syntactic structure for the free word order languages.

The number of parameters for estimating the probabilities of the dependency trees may be infinite owing to the recursiveness of languages. So it is necessary to decompose the tree into smaller fragments. Here, we break down the tree into a set of dependency relations, or arcs, to reduce the number of parameters to estimate.

And the useful contextual information for the parsing model is investigated. Generally, the parameters include the contextual features to maximize the discriminative power of them. Discriminative power is ability to distinguish syntactic structures that are correct or wrong. Higher discriminative power of a tree fragment will be more helpful for syntactic disambiguation. The contextual information, such as distance between depending words, neighboring words, is looked into and motivation of using the information in the parsing model was investigated.

This thesis doesn't discuss the dependency grammar for Korean itself. The methodology used in this thesis just counts the number of dependency relations from a dependency-annotated Treebank. It doesn't use explicit grammar at all. And this thesis doesn't discuss syntactic disambiguation using other linguistic resources except the parsed corpora. There-

fore it can be applied to any other languages, which are linguistically similar to Korean, if dependency-parsed corpora exist for the languages.

One of the important points of this work is to show that even some features related to word order are useful for parsing free word order language. Furthermore, this work shows the way to parameterize the feature affects the performance of the parsing model and the parameters designed for the variable order languages work better than those for the fixed order languages in parsing Korean. That is to say, the probabilistic model of the natural language depends on the characteristic of language. This thesis provides the experimental support for the claim. The parser for the statistical model is constructed and the performance of the parser is compared with the parsers using other models on the identical training and testing data.

1.3 Overview

The outline of this dissertation is as follows:

Chapter 2 introduces some fundamental knowledge related to this dissertation. Dependency grammar, Korean grammar, and parsing Korean with dependency grammar are explained in order.

Chapter 3 lists previous works on the parsing Korean. First history on parsing the Korean language is introduced briefly. Then the statistical language or parsing models are explained in detail. Most of them are statistical models for parsing Korean with dependency grammar, but some of them deal with the statistical parsing model for Japanese, or phrase structural model for parsing Korean.

Chapter 4 considers the parameterizations for the dependency parsing problem. We define the concept of *Event*, *Prediction*, and *Context* used in the proposed parsing model. We list the features used for the *Context* and describe the motivation for using them as the contextual information. And we compare our parameterization with others.

Chapter 5 describes the statistical parsing model. It models the prediction and context given in the previous chapter. The explanation on the parser and the parameter estimation

follows. The experimental result shows the performance of the parsing model and comparison with other parsers. The effect of each feature used in the proposed parsing model is discussed. Closer look into the experimental results is given to talk over the effect of newly added features.

Chapter 6 discusses the result from Chapter 5. Some discussions on unlexicalized parsing and parsing with clustered word are shown in this chapter. The word clustering was done according to their preference on the modification distance. The approaches described here are tested with the same testing data from Chapter 5, and compared with the result from the previous chapter.

Chapter 7 concludes the dissertation. Contributions of the dissertation and some thoughts on future works are given.

Chapter 2

Preliminaries

2.1 Dependency Grammar

In *Éléments de syntaxe structurale*, Lucien Tesnière founds a unique syntactic theory named *dependency theory* (Seo 1998). *Dependency* is an asymmetrical relation between a head and a dependent (Kruijff 2002). Heads and dependents are related immediately (e.g. there are no nonterminals). Dependency grammar is a set of rules that describes the dependencies. The observation that drives dependency grammar is simple: Every word (dependent) depends on another word (head), except one word, which is the root of the sentence (Debusmann 2000). With dependency grammar, a sentence *A man sleeps* can be analyzed as

a depends on *man*

man depends on *sleep*

sleeps depends on nothing (i.e. is the root of the sentence)

Or it can be expressed as

a modifies *man*

man is the subject of *sleep*

sleeps is the matrix verb of the sentence

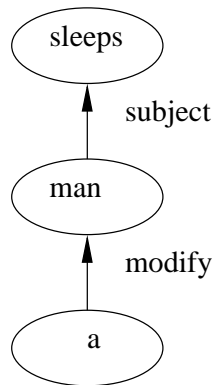


Figure 2.1: A dependency tree of the sentence *a man sleeps*. *sleeps* is the root of the tree, or the sentence. Sometimes, the dependency arc labels (e.g. *subject*, *modify*) are omitted

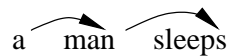


Figure 2.2: Another way to express the dependency structure of the sentence *a man sleeps*.

A dependency structure is a collection of dependencies for a sentence. The dependency structure of the sentence can be expressed with a tree, as in Figure (2.1) and Figure (2.2). A more complex dependency structure is shown in Figure (2.3). Unlike phrase structure grammars, the dependency grammar does not divide the sentence up into constituents, but only identifies the grammatical relations between words. This is advantageous in language where the order of words is variable (Covington 1990).

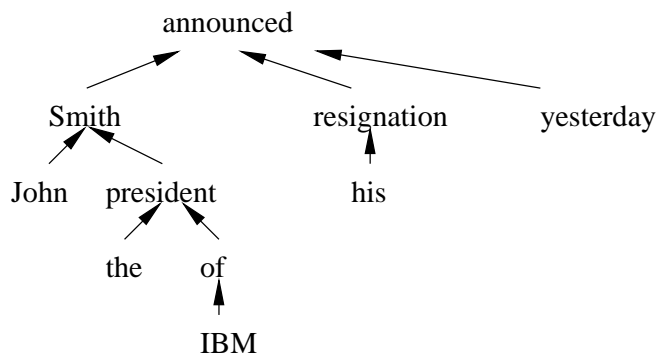


Figure 2.3: Dependency structure for the sentence *John Smith, the president of IBM announced his resignation yesterday.*

2.2 A Brief Introduction to Korean Grammar

Korean has two characteristics that have to be taken into account: agglutinative morphology and rather free word order¹ with explicit case marking (Lee 1997a).

Morphology

Korean is an agglutinative language, in which a word² is in a composition of more than one morpheme, in general. There are two types of morpheme, which are a content morpheme and a functional morpheme (Lee and Bae 1987). A content morpheme contains the meaning of the word. Nouns, verbs, adjectives, and adverbs belong to this category. A functional morpheme plays a role as a grammatical information marker, which indicates a grammatical role, tense, modality, voice, etc. of the word. In an example sentence

(2.1)	유진이	쇼를	보았다.
	<i>Eugene-i</i>	<i>show-reul</i>	<i>bo-at-da</i>
	Eugene-NOM	show-ACC	watch-PAST-END

¹*Free word order* is a traditional term that should not be taken literally. But many researches retain the term for its conciseness (Bien and Szpakowics 1982).

²The exact term for the word is *eojeol* in Korean

Eugene saw a show.

, the first word consists of two morphemes, 유진 (*Eugene*) and 이 (*-i*)³. 유진 (*Eugene*) is a content morpheme and 이 (*-i*) is a functional morpheme which is a nominative case marker. The second word consists of 쇼 (*show*) and 를 (*-reul*) which are a content morpheme and an accusative case marker. The last word consists of three morphemes; 보 (*bo*), 았 (*-at*), and 다 (*-da*). 보 (*bo*) is a content morpheme, which is a verb and means *watch*, while two other morphemes are functional morphemes. 았 (*-at*) is a past tense marker and 다 (*-da*) is a final ending marker.

Syntax

The order of words is relatively less fixed in Korean compared to the fixed-order languages such as English. The grammatical information conveyed by a functional morpheme makes a word order be free. For example, a subject normally precedes an object in English, while a subject can precede an object or vice versa in Korean. Not word order but the functional morpheme in a word decides whether a word is a subject or an object.

The following Korean sentence consists of 4 words.

(2.2) 어제 유진이 쇼를 보았다.
eoje *Eugene-i* *show-reul* *bo-at-da*.
yesterday Eugene-NOM a show-ACC watch-PAST-END
Eugene watched a show yesterday.

The sentence can be rewritten to the following due to the weak word order restriction in Korean.

(2.3) 어제 쇼를 유진이 보았다.
eoje *show-reul* *Eugene-i* *bo-at-da*.
yesterday a show-ACC Eugene-NOM watch-PAST-END
Eugene watched a show yesterday.

Though the subject and the object are exchanged their position, the sentence preserves its meaning. Because of this property of Korean, dependency grammar is widely used for

³Hangeul (Korean Alphabet) romanization follows (of Culture & Tourism 2000)

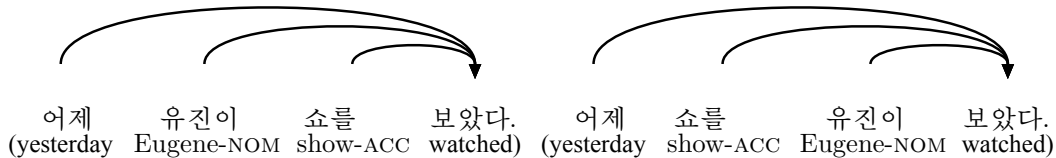


Figure 2.4: Dependency trees for Korean sentences which have identical meaning, *Eugene watched a show yesterday*.

analyzing syntactic structure of the Korean language. Figure 2.4 shows dependency structure trees for the Korean sentences shown above.

2.3 Parsing Korean with Dependency Grammar

Because the dependency grammar easily handles free-word order, discontinuous constituents, and constituent ellipsis, it has been popular in parsing Korean (Ra 1994). Some issues for applying the dependency syntax theory to Korean are introduced in this section.

2.3.1 Word Modeling

A parsing unit for dependency grammar is a word in English⁴. For Korean, a morpheme or a word can be mapped to a parsing unit. For earlier researches on the dependency grammar for the Korean language, such as Kwon and Choi (1992), a morpheme was used as a parsing unit. However, Seo (1993) proposed dependency parsing based on word-unit, instead of the morpheme-unit, and this approach has been widely used until now. Since a Korean word contains more than one morpheme, the syntactic category of a word has to be decided based on the morphemes that consists the word, if a word is used as a parsing unit. He defines that a word has two syntactic categories; the left category and the right category. The content morpheme in the word is used as the left category of the word, while the functional

⁴See Figure 2.1 and 2.3 for examples

Category of Head	Category of Dependent	Dependency Relation
noun	noun, numeral, pronoun, ...	modify
numeral	pronoun, demonstrative adnoun, ...	modify
attributive adnoun	attributive adverb	adjunct
...
numeral adnoun	attributive adnoun	modify
verb	noun, pronoun, numeral, case-marker	case relation
verb	adverbial-ending	adjunct

Table 2.1: A part of the Korean dependency grammar

morpheme in the word is used as the right category of the word. For example, the following words

(2.4) 유진이 잔다.
Eugene-i ja-nda.
 Eugene-NOM sleep-END
 Eugene sleeps.

have two categories for each. For the first word 유진이 (*Eugene-i*),

Pronoun is the left category
case marker is the right category

and for the second word 잔다 (*ja-nda*),

verb is the left category
final ending is the right category

The syntactic category of a word varies according to its role in a grammatical rule. If the word is used as a modifier, the left category is set to the category of the whole word. If the word is used as a head, the right category is used.

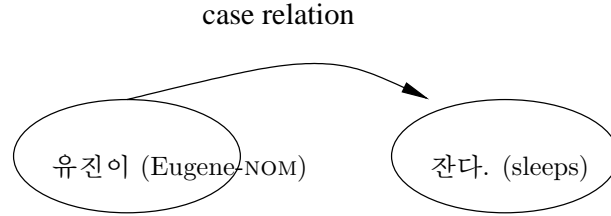


Figure 2.5: Dependency relation between 유진이 (*Eugene-i*; Eugene-NOM) and 잔다 (*ja-nda*; sleeps). The right category (case-marker) of 유진이 (*Eugene-i*) and the left category (verb) of 잔다 (*ja-nda*) make two words have dependency relation.

2.3.2 Dependency Rule

A dependency rule defines the relation between two words⁵. Table 2.1 is a sample of the dependency grammar from Kim, Kim, and Seo (1993), which is a set of dependency rules. For example, the last rule of the grammar says the word with an adverbial right category can modify the word categorized as a verb, and their dependency relation is adjunct. The rules are used for analyzing sentences. For instance, the two words in the example 2.3.1 are analyzed by the dependency rule

case-marker → *verb* (6th line in Table 2.1),

which means the right syntactic category of the first word and the left syntactic category of the second word take part in as Figure 2.5.

2.3.3 Constraints based on the Structural Properties of Korean

Although the dependency theory has many advantages in analyzing the variable word order languages such as Korean, it has also drawbacks. One of the drawbacks of parsing with dependency rules is allowing too many incorrect dependency relations. To reduce the number

⁵Or two syntactic categories more accurately, because a word is represented with a syntactic category of the word.

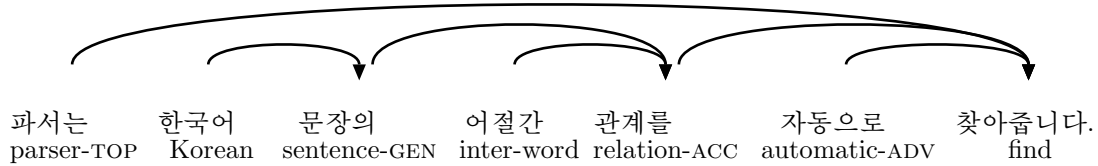


Figure 2.6: All dependents precede their heads

of incorrect dependency relation, various dependency constraints designed for Korean are given to the parser. Some of typical constraints used frequently in analyzing Korean are followings (Seo 1993; Ra 1994)⁶:

All dependents precede their heads

Generally, all dependent words precede their heads in Korean sentence. Figure 2.6 is a dependency structure for the sentence:

- (2.5) 파서는 한국어 문장의 어절간 관계를 자동으로
parser-neun hangugeo munjang-ui eojeolgan gwangye-reul jadong-euro
 parser-TOP Korean sentence-GEN inter-word relation-ACC automatic-ADV
 찾아줍니다.
chajajumnida.
 find.

The parser finds out inter-word relations in a Korean sentence automatically.

You can see all the heads of the words in the sentence are placed after their modifiers. There are few exceptional sentences violating this property. So most of dependency parsers use this constraint.

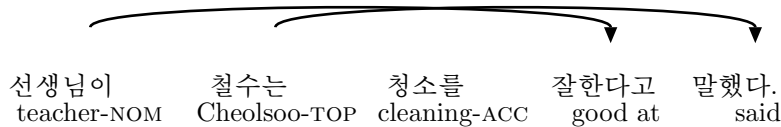


Figure 2.7: Wrongly analyzed sentence with crossing dependencies.

Dependencies don't cross

The Korean language does not allow the crossing dependency relation in general. The crossing relations change the meaning of the original sentences and convey the wrong meaning (Lee 2002). Let's consider the following sentence:

- (2.6) 선생님_i 철수_는 청소_를 잘_하다고 말_했다.
seonsaengnim-i Cheolsoo-neun chungso-reul jalhandago malhaessda.
 teacher-NOM Cheolsoo-TOP cleaning-ACC good at said
 The teacher said Cheolsoo is good at cleaning.

Figure 2.7 is wrongly analyzed dependency structure. If the sentence is analyzed as Figure 2.7, then the meaning of the sentence changes to *The Cheolsoo said the teacher is good at cleaning*. To preserve the original meaning of the sentence, the dependency structure containing crossing relations are avoided.

Most of dependency parsers – even statistical parsers – for Korean use these elementary constraints for efficiency.

⁶These characteristics of dependency relations are similar for that of Japanese (Sekine, Uchimoto, and Isahara 2000), which has similar syntax with Korean

Chapter 3

Previous Work

This chapter gives a broad overview of the researches on parsing of Korean. We first give a brief history of the field and describe literatures directly related to our work after it.

3.1 A Brief History of Parsing for Korean

Parsing with Unification Grammar Framework for Korean

As Yoon (1993) states, research for the Korean language parsing has begun from 1980s. Korean parsers based on the Marcus parser was introduced in early 1980s and used for the simple Korean-English machine translation system.

In mid 80s, the syntactic analysis for Korean has been tried with the unification grammar framework. Geum and Kim (1988), Jung, Kim, Lee, Chun, and Park (1989), Yang (1990) and Seo and Kim (1990) used HPSG for the Korean language parsing. Yoon and Kim (1989) used LFG for developing a Korean parser.

Increase of Interest in Dependency Parsing

Dependency grammar has been widely used for parsing Korean from the 90s. Dependency analysis thought to be an appropriate method for analyzing variable word order languages like Korean, because of its word-based characteristics. Yoon (1990), Woo (1992), Yoon

and Kim (1992), Kwon and Choi (1992), and Seo (1993) started applying the dependency grammar to the Korean analysis.

Parsing with Probabilities

Until the mid 90s, people tried to build syntactic constraints for the Korean language. The constraints were inscribed through the unification frameworks or encoded in the parsers to remove overly generated parses. Meanwhile, statistical approaches for natural language processing became popular under the influence of the engineering and acoustic communities (Charniak 1993). This affects the parsing methodologies for the Korean language, too.

Kim (1994) defined a probabilistic dependency grammar for Korean. However, little literature on probabilistic parsing was published right after the work, because Treebank data for training the model were unavailable. A few years from then, researchers at Sogang University (Kim and Seo 1997; Kim, Kim, and Seo 1997) proposed statistical parsing models and trained them with their own small sized training corpus (498 sentences).

In 1997, Center for Korean Language Engineering released a Treebank¹, which was labeled with a Korean phrase structure grammar (Lee, Kim, Chang, Choi, and Kim 1996). Many researchers used this Treebank to train their parsing model. Lee used this corpus and proposed probabilistic parsing models for Korean (Lee, Kim, and Kim 1996; Lee 1997a; Lee, Kim, and Kim 1998). Seo, Nam, and Choi (1999) built a statistical dependency parsing model and trained it with the dependency tagged version of the Treebank. Kwak, Park, Hwang, Chung, Lee, and Rim (2003) designed probabilistic generalized-LR (GLR) parser trained with the Treebank. They converted the KAIST Treebank in the form of their own grammar (Park, Hwang, and Rim 1999), which is the feature-based phrase structure grammar for Korean.

¹The Treebank is a part of the evaluation release of KOREA NATIONAL LANGUAGE INFORMATION BASE. Funded by the Ministry of Science & Technology and the Ministry of Culture of Korea, the information base was constructed to improve Korean language processing technology and to promote Korean software industry. The size of the Treebank in the evaluation release is 10,000 sentences, while the full version of the Treebank consists of more than 30,000 sentences. It is usually known as the KAIST Treebank or the KAIST language resources (Choi 2001).

Hybrid Approaches to Parsing

Another prominent trend from the 90s was hybrid approach, which combines statistical method with rule-based method. Yoon (1997) uses lexical co-occurrence statistics from raw corpora with some constraint rules in his dependency parser. Jung, Park, Ra, and Yoon (2001) used similar statistical data with simple heuristic rules to implement a dependency parser for Korean. Kim, Park, Ra, and Yoon (2002) suggests parsing method for Korean that uses the statistical data and sentence segmentation. Kim, Kang, and Lee (2001) used valency information and structural preference rule with statistical information from the KAIST Treebank for resolving ambiguities in dependency parsing.

3.2 Related Work

This thesis proposes a new statistical dependency parsing model for the Korean language. This section describes various statistical parsing model related to our model in detail.

(Kim 1994)

Kim (1994) illustrates a bridging effort between dependency grammar for Korean syntax and statistical language modeling. He designed a statistical language model based on the probability of a dependency relation:

$$P(t_i|t_j) = \lambda_1 P_1(t_i|t_j, d) + \lambda_2 P_2(t_i|t_j, d) \quad (3.1)$$

$$\text{where } d = \begin{cases} 1 & \text{if } j - i = 1 \\ 2 & \text{if } j - i > 1 \end{cases}$$

t_i is the part-of-speech tag of the i th-word in a sentence. For alleviating data sparseness problem, P_2 is used with P_1 . P_2 is a back-off probability for P_1 . That is,

$$P_1(t_i|t_j, d) = \frac{C(t_i, t_j, d)}{\sum_{x < j} C(t_x, t_j, d)} \quad P_2(t_i|t_j, d) = \frac{C(ft_i, ct_j, d)}{\sum_{x < j} C(ft_x, ct_j, d)}$$

while ft_x and ct_x are part-of-speech tags of functional and content morpheme of the x -th word. He recognizes the syntactic structure of Korean prefers a dependency relation between adjacent words and include the adjacency measure d in his model. The parsing model was trained and tested with 1088 sentences collected from various domains. 20% of the set was used as testing data. It gives around 80% of arc-based precision.

(Collins 1996)

Collins (1996) describes a statistical English parser based on bigram lexical dependencies. Given a sentence S , the statistical model chunks baseNPs and sets dependencies D between words or the baseNPs.

$$P(T|S) = P(B, D|S) = P(B|S) \cdot P(D|S, B) \quad (3.2)$$

where T is a parse tree and B is a set of baseNPs. The constituent structures of Penn Treebank were converted to dependency structures for training the model. It assigns a probability to the dependency relation² from i th word w_i to j th word w_j , with relationship R as follows :

$$P(AF(i) = j, R|S, B) = \frac{P(R|w_i, w_j, \Delta)}{\sum_{k \neq i, r \in REL} P(r|w_i, w_k, \Delta)} \quad (3.3)$$

REL is a set of relationship of dependency. He used the distance variable Δ to consider additional context. Back-off is used to smooth the above probabilities. The parser trained on Penn Wall Street Journal corpus performs about 85.5 % of labeled precision and recall for the heldout testing data.

(Kim and Seo 1997)

Kim and Seo (1997) modified the parser of Collins (1996) to consider the characteristics of Korean. A right to left chart parsing mechanism (Kim 1993) is used. They add some

²He uses the notation $AF(i) = (j, R)$ to state the dependency relation. AF stands for *arrow from*.

simple heuristics to control the number of analyzed result and improve efficiency. In the probabilistic parsing model, a word unit is altered into a representative morpheme to consider morphology of the Korean language.

$$P(AF(i) = j, R|S, B) = \frac{P(R|cm_i, fm_i, cm_j, fm_j, d)}{\sum_{k \neq i, r \in REL} P(r|cm_i, fm_i, cm_k, fm_k, d)} \quad (3.4)$$

where cm_i and fm_i are a content morpheme and a functional morpheme of w_i . The distance measure d is just a simple surface distance between two depending words. The parser is trained with 494 sentences and tested on 100 sentences. It performs at arc-based recall of 78% with 42% of the sentence-level accuracy.

(Kim, Kim, and Seo 1997)

Kim, Kim, and Seo (1997) discuss the drawback of the distance measure Kim and Seo (1997) has used. Instead of the surface distance, they suggest to use a phrasal distance for certain types of dependency relation.

$$d = \begin{cases} d_p & \text{if } (d_s - d_p) > \epsilon \\ d_s & \text{otherwise} \end{cases}$$

where d_p and d_s are the phrasal distance and the surface distance. They compared the experimental result of this modified model with that of the (Kim and Seo 1997). They insist the modified model shows better performance for longer input sentences. However, the parser of Kim and Seo (1997) achieves much higher accuracy in short sentences and it also performs better than the parser considering phrasal distance even in parsing longer sentences, if the parser outputs the most probable parse.

(Lee 1997a)

Lee (1997a) investigates two problems in her dissertation : Defining Korean syntax and building a probabilistic language model for Korean. The KAIST Treebank is constructed based on the Korean syntax defined here. She publishes the first statistical parser trained with the Treebank.

Her basic language model considers outside context information to the probabilistic context-free grammar (PCFG) language model.

$$P(T, S) = \prod_{rule \in T} P(rule | t_{ol}, t_{or}) \quad (3.5)$$

where t_{ol} and t_{or} are the left and the right outside context of the constitute covered by the *rule* (e.g. Part-of-speech tags of the outside of both ends of the coverage which the *rule* spans). She extends the basic model to include lexical co-occurrence probability and a distance variable representing surface distance between two heads of the child constituents. Simply the extended model is

$$P(T, S) = \prod_{rule \in T} P(rule | outer\ context) \cdot P(lexical\ co - occurrence, distance) \quad (3.6)$$

The model is trained with 30,086 sentences of the KAIST Treebank. The best results on its test set are 84.83%/84.15% labeled precision/recall rate. By comparing her model with the parsing model for English (i.e. models of Charniak (1995) and Collins (1996)) empirically, she concludes that a language model cannot be completely independent of specific language nor a syntactic representation. That is, it is important to consider the language specific features when building a statistical language model.

(Seo, Nam, and Choi 1999)

Most of the parsing models introduced before uses parameters derived from the word order, such as distance measure. Seo, Nam, and Choi (1999) insists such parameters may not play any role in disambiguating the structure of the variable-word-order languages. They devise a language model for the variable-word-order languages based on dependency grammar that considers ascending dependency. The ascending dependency, which is not the word order based parameter, is the relationship between a word and its ancestors in the hierarchy of the dependency relations. While k -ascendants of w_i is defined as a list of w_i 's ascendants whose orders are less than k , and written as H_i^k , the language model is

$$P(T|S) = \prod_i P(w_i|H_i^k) \quad (3.7)$$

Considering the morphological structure of a Korean word, he replaces the word parameter w_i to its functional word f_i . Also, H_i^k , is replaced by the list of the content words CH_i^k , for the same reason.

$$P(T|S) = \prod_i P(w_i|H_i^k) \approx \prod_i P(f_i|CH_i^k) \quad (3.8)$$

He uses the KAIST Treebank to evaluate his language model. 20,000 sentences are used as training data and 210 sentences are used as heldout testing data. The part-of-speech tags are used instead of the lexical words to avoid data sparseness problem. By changing the size of k , he finds out 2-ascendant model works most effectively in the corpus he used. It achieves 85.7% accuracy at recovering dependency relations. It results better than the models of Eisner (1998) and Collins (1996) in parsing Korean.

(Kanayama, Torisawa, Mitsuichi, and Tsujii 1999; Kanayama, Torisawa, Mitsuichi, and Tsujii 2000)

These works are not for parsing the Korean language, but for the Japanese language. Both works share an identical idea but implement the idea differently; Kanayama, Torisawa, Mitsuichi, and Tsujii (1999) uses maximum likelihood estimation, while Kanayama, Torisawa, Mitsuichi, and Tsujii (2000) uses ChoiceMaker Maximum Entropy Estimator.

Both works utilized handcrafted HPSG for dependency analysis of Japanese. HPSG is used to restrict the candidates for modification to syntactically valid ones and their probabilistic models chooses the actual head among at most three alternative head candidates: the nearest, the second nearest, and the farthest candidates from a certain modifier.

The reason of considering only three alternative head candidates is their observation on the statistical distribution of heads in Japanese sentences. According to their investigation, 98% of *bunsetsus* (the phrasal units in Japanese, which can be mapped to words

in Korean) modify one of those three candidates. This restriction results the design of the *triplet/quadruplet model*.

$$P(t_i \rightarrow t_j | S) = \begin{cases} 1 & \text{if } noc(i) = 1 \\ \epsilon & \text{if } t_j \notin \{cd(i, 1), cd(i, 2), cd(i, last)\} \\ P(k | t_i, cd(i, 1), cd(i, last)) & \\ \quad \text{if } noc(i) = 2 \ \& \ t_j \in \{cd(i, 1), cd(i, last)\} \\ P(k | t_i, cd(i, 1), cd(i, 2), cd(i, last)) & \\ \quad \text{if } noc(i) \geq 3 \ \& \ t_j \in \{cd(i, 1), cd(i, 2), cd(i, last)\} \end{cases} \quad (3.9)$$

$cd(i, y)$ is the part-of-speech tag of the y th head candidate of w_i and $noc(i)$ is the number of head candidate for w_i .

These models seem to work well for Japanese, however, it is doubtful that the parsing models can be applied to other languages well. The parsing models are restricted to consider only three head candidates at most, based on the statistics from Japanese corpora. So they may fit for Japanese parsing but would cause problems for parsing other languages. And these approaches require handcrafted grammars that usually demand excessive manual labors. These features can be obstacles when someone uses these models to develop a new parser for other languages.

Chapter 4

Parameterization for Dependency Parsing

In this chapter, we discuss the parameterization of a parse tree for probabilistic dependency parsing. The parameterization of a parse tree is the choice of how to break down a parse tree (Collins 1999). We need the probabilities of parse trees for statistical syntactic disambiguation. However, the creative power of a language is infinite; the number of sentence types are infinite, and the length of a sentence can be infinite. These make the number of parse tree be infinite. Therefore it is almost impossible to get the probability of a certain parse tree without breaking down the parse tree to smaller segments, or events. In other words, the probability of a parse tree for a given sentence is estimated through the probability product of events for given context information from the sentence.

$$P(T|S) = \prod_i P(Event_i|S) = \prod_i P(Prediction_i|Context_i) \quad (4.1)$$

The probability of an event for a given sentence is measured by the probability of a decision made in a given context. The parameterization, including selections of $Event_i$, $Prediction_i$ and $Context_i$ is crucial for syntactic disambiguation. The choice of using a particular parameterization should be based on its ability to discriminate between different

parses.

4.1 Selection of $Event_i$

A dependency parse tree is a set of dependency relations. A parse tree contains $|S| - 1$ number of dependency relations, where $|S|$ is a number of words in the sentence S . Most of the previous statistical dependency parsing models define an event as forming a dependency relation (Kim 1994; Collins 1996; Kim and Seo 1997; Kim, Kim, and Seo 1997; Seo, Nam, and Choi 1999; Kanayama, Torisawa, Mitsuichi, and Tsujii 1999; Kanayama, Torisawa, Mitsuichi, and Tsujii 2000). We also follow the definition: dep_i , the dependency relation from the i -th word in a sentence is used as the $Event_i$

$$P(T|S) = \prod_i P(Event_i|S) = \prod_i P(dep_i|S) \quad (4.2)$$

4.2 Selection of $Prediction_i$

Our probabilistic parsing model predicts *the existence of a dependency relation between two words* and *the length of the dependency relation* in a certain circumstances, or a context, and the probability of the prediction is the probability of the event, or dependency relation.

$$P(dep_i|S) = P(head(w_i) = w_j, length(w_i) = d|Context_i) \quad (4.3)$$

w_i and w_j are words, d is the length of the dependency relation, $head(x)$ is the head word of the word x and $length(x)$ is the length of the dependency relation from the word x .

4.3 Selection of $Context_i$

$Context_i$ is contextual information for $Decision_i$, or a basis for the decision. There are many ways to set $Context_i$. The simplest way may be using the part-of-speeches of two depending words, w_i and w_j as $Context_i$. Or more complex linguistic features can be included in $Context_i$. The related researches shown in the previous chapter differs mostly in what to

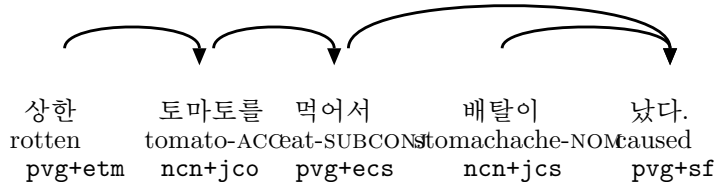


Figure 4.1: The dependency tree with part-of-speech tags for Example (4.5)

consider in $Context_i$. Deciding $Context_i$ is one of crucial point in designing a probabilistic model for parsing.

Many kinds of $Context_i$ will be considered in this section. We will compare each $Context_i$ to others, and give some motivation of using the each contextual information.

4.3.1 Part-of-Speech Tags of Depending Words

The simplest way to consider the context is looking two part-of-speeches: one is the right category of a modifier and the other is the left category of a modifyee.

$$P(\text{head}(w_i) = w_j \mid \text{length}(w_i) = d \mid ft_i \ ct_j) \quad (4.4)$$

where ft_i is the right category, or functional tag, of the word w_i , and ct_i is the left category, or content tag, of the word w_i .

Assume we are deciding whether there is a dependency relation between two words 상한 (*sangha-n*; rotten) and 먹어서 (*meog-eoseo*; eat-SUBCONJ) in the following sentence.

(4.5) 상한 토마토를 먹어서 배탈이 났다.
sangha-n *tomato-reul* *meog-eoseo* *baetal-NOM* *na-tda.*
rotten tomato-ACC eat-SUBCONJ stomachache-NOM caused
Eating a rotten tomato caused a stomachache.

Then the probability of having the dependency relation between *jco* and *pvg* when its length is 1 is used to decide the dependency relation between the two words.

$$P(\text{head}(\text{sangha-n}) = \text{meog-eoseo} \mid \text{length}(\text{sangha-n}) = 1 \mid \text{jco pvg}) \quad (4.6)$$

The context provides the syntactic categories of two words (Figure 4.1 shows the part-of-speeches of the words). The syntactic category is very important in deciding whether the two words can be joined by a dependency relation; for example, the word with accusative case marker is not allowed to have dependency relation with an adjective.

And the two part-of-speech tags can give a clue for deciding the modification length from the modifying word. The two part-of-speech `jco` and `pvg` are likely to be placed close to each other, which means the length of the dependency relation is short, because most of arguments used to lie close to their predicate.

4.3.2 Lexical Bigram Dependency

Lexical form of a word contains very important information, which is a *meaning* of the word, and relevance between two lexical forms is used to get selectional preference between two depending words. The selectional preference is the preference of the combinations of words that can co-occur. For example, the word pair

<토마토를 (*tomato-reul*;tomato-ACC), 먹어서 (*meog-eoseo*; eat-SUBCONJ)>

is much preferred to the another word pair

<토마토를 (*tomato-reul*;tomato-ACC), 났다 (*nat-da*; budded) >

, and this preference helps syntactic disambiguation process.

If the lexical information is used as the *Context_i*, the probability of a dependency relation becomes as following:

$$P(\text{head}(w_i) = w_j \mid \text{length}(w_i) = d \mid w_i w_j) \quad (4.7)$$

It measures the lexical association, or selectional preference between two words w_i and w_j

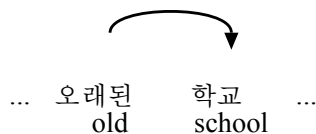


Figure 4.2: A dependency relation between 오래된 (*oraedoe-n*; old) and 학교 (*hakgyo*; school).

when they lie d words away to each other. The probabilities between the word pairs showed above are:

$$P(\text{head}(\text{tomato-reul}) = \text{meog-eoseo} \mid \text{length}(\text{tomato-reul}) = 1 \mid \text{toamto-reul meog-eoseo}) \quad (4.8)$$

and

$$P(\text{head}(\text{tomato-reul}) = \text{na-tda} \mid \text{length}(\text{tomato-reul}) = 3 \mid \text{toamto-reul na-tda.}) \quad (4.9)$$

, and these probabilities are used in modifyee selection.

The problem of using the lexical information for the context is data sparseness. It is hard to obtain reliable probability of lexical pairs from training corpus, because the number of lexical item is very large compared with the size of training corpus.

4.3.3 Surrounding Words Information

Surrounding words around the two depending words affect the $Prediction_i$. The dependency probability between two words can be more elaborately estimated when conditioned on the words that surround the dependency relation, as well as the two words in the dependency relation.

$$P(\text{head}(w_i) = w_j \mid \text{length}(w_i) = d \mid w_i \ w_j \ \Phi_i \ \Phi_j) \quad (4.10)$$

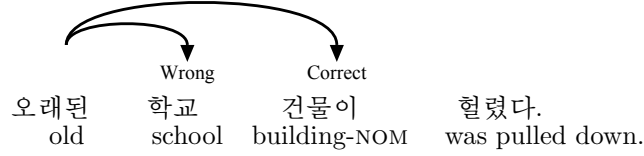


Figure 4.3: A wrong and a correct head for the word 오래된 (*oraedoe-n*; old), in the sentence meaning *The old school building was pulled down*.

where Φ_x is the information of the surrounding words of w_x .

For example, the likelihood of the dependency relation between the determinative word and the noun may vary with the right context of the noun. Take the case in Figure 4.2. There is a dependency relation between the determinative word 오래된 (*oraedoe-n*; old) and the noun 학교 (*hakgyo*; school). The right context of the word 학교 (*hakgyo*) affects the likelihood of this dependency relation; if the right context is another noun, the likelihood of the relation may decrease because the noun next to 학교 (*hakgyo*) can be chunked with the 학교 (*hakgyo*) to build a noun phrase. In this case, the determinative word 오래된 (*oraedoe-n*) modifies the head of the noun phrase, which is the next word of 학교 (*hakgyo*). The sentence

- (4.11) 오래된 학교 건물이 헐렸다.
 oraedoe-n hakyo geonmul-i heolly-eosda.
 old school building-NOM was pulled down.
 The old school building was pulled down.

is the case. The noun phrase made by 학교 (*hakgyo*) and the next word 건물이 (*geonmul-i*; building-NOM) is modified by the modifier 오래된 (*oraedoe-n*). Of course, the modifier have a dependency relation with the noun 건물이 (*geonmul-i*), which is the head of the noun phrase (Figure 4.3).

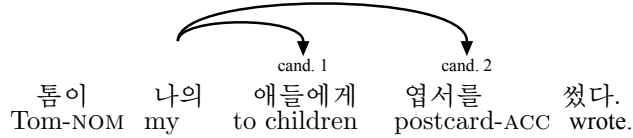


Figure 4.4: The sentence means *Tom wrote a postcard to my children.* The word *나의* (*na-ui*) has two alternative modifyee candidates

4.4 Splitting the $Prediction_i$

We split the $Prediction_i$ into two parts: the prediction for distance of modification ($length(w_i) = dist$), and the prediction for the word dependency relation ($head(w_i) = w_j$). The reason for this splitting is the data sparseness problem. Dividing the prediction into two sub-predictions helps alleviating the problem by allowing us to use two different contexts to each prediction. Each context we used is selected according to our observation.

4.4.1 Context for Modification Distance Prediction

We have observed that a human being can predict how far a word he (or she) is reading will modify although the sentence has not been completely read. For example, read the following sentence.

- (4.12) 톰이 나의 애들에게 엽서를 썼다.
Tom-i *na-ui* *aideul-ege* *yeobseo-leul* *sseosda.*
 Tom-NOM my to children postcard-ACC wrote.
 Tom wrote a postcard to my children.

At the moment we read the word *나의* (*na-ui* ; *my*), we already know that the word will modify the next word by instinct. Another example:

(4.13)	2000년이	되면	아태지역은	EC보다	큰
	<i>2000nyeon-i</i>	<i>doe-myeon</i>	<i>ataejijeog-eun</i>	<i>EC-boda</i>	<i>keun</i>
	year 2000-CMPL	become-SUBCNJ	Asian-Pacific region-TOP	EC-COMP	large

...
...

In the year 2000, the Asian-Pacific region ... larger than EC.

Reading the word *되면* (*doe-myeon*), we can know that it modifies the word farther on, even though we don't read further. How can we know how far those words modify? How can we guess the lengths of dependency relation from those words?

We predict the length of dependency relation from a certain word by knowing the syntactic characteristic of the word. That means knowing the syntactic characteristic¹ of the word makes the head selection for the word easier. Revisit the sentence (4.12). As Figure 4.4 shows, the word *나의* (*na-ui*; my), which is a noun modifier, has two alternative noun head candidates : *애들에게* (*aideul-ege*; to children) and *엽서를* (*yeobseo-leul*; postcard-ACC). Here, the first candidate is the correct head for the modifier. It is well known to native Korean users that the word ends with the morpheme *의* (*-ui*; genitive postposition) usually modifies the right next word. In other words, the word ends with the genitive marker *의* (*-ui*) prefers modification distance of 1 generally.

Knowing the surrounding context of the word makes the expectation more accurate. Consider the following sentence.

(4.14)	툼이	나의	아픈	애들에게	엽서를	썼다.
	<i>Tom-i</i>	<i>na-ui</i>	<i>apeun</i>	<i>aideul-ege</i>	<i>yeobseo-leul</i>	<i>sseosda.</i>
	Tom-NOM	my	sick	to children	postcard-ACC	wrote.

Tom wrote a postcard to my sick children.

The word *아픈* (*apeun*; sick) is added next to the *나의* (*na-ui*; my). This addition makes *나의* (*na-ui*) not modify the next word, but modify the two words apart from it (Figure 4.5). In above sentence, the decision of the modification distance may be inaccurate without the knowledge of the next word *아픈* (*apeun*). This suggests that the contextual information of the word is helpful for reflecting the syntactic preference of the word. Some rule-based

¹Here, the syntactic characteristic solely refers to the preference on modification distance for a word.

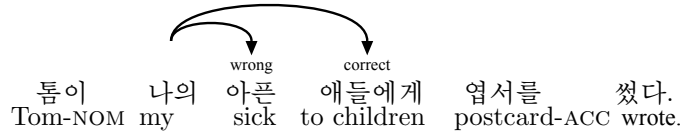


Figure 4.5: The sentence means *Tom wrote a postcard to my sick children*. The word ends with 의 (-*ui*) doesn't modify the next word because of the inserted modifier 의 (-*ui*)아픈 (*apeun*).

or heuristic-based parsers encoded this preference into a rule for syntactic disambiguation (Ryu, Lee, and Lee 1996). Let's see another sentence:

- (4.15) 공연은 완전히 실패한 것으로 드러났다.
 gongyeon-eun *wanjeonhi* *silpaecha-n* *gut-euro* *deurona-tda*.
 show-SBJ completely failed that was revealed
- It was revealed that the show was completely failed.

The adverb 완전히 (*wanjeonhi*; completely) has two alternative head candidates in this sentence. They are 실패한 (*silpaecha-n*; failed) and 드러났다 (*derona-tda*; was revealed), and the former is the correct head of the adverb (Figure 4.6). Finding the correct head is tough in this case, although we consider lexical or semantic information, because the lexical or semantic preference of the adverb 완전히 (*wanjeonhi*) to both modifyee candidates are similar².

However, we observed that even a word like that has the tendency to have a fixed modification distance, or dependency length, in a certain context. For instance, the adverb

²Two verb phrases <완전히, 실패한> (*wanjeonhi, silpaecha-n*; completely failed) and <완전히, 드러났다> (*wanjeonhi, derona-tda*; completely revealed) are fluently read to native Korean speakers. None of the pairs are awkward. Being statistically measured, the <완전히, 드러났다> (*wanjeonhi, derona-tda*; completely revealed) pair was little more likely to be co-related than the <완전히, 실패한> (*wanjeonhi, silpaecha-n*; completely failed) pair (41.9% vs. 50.76%).

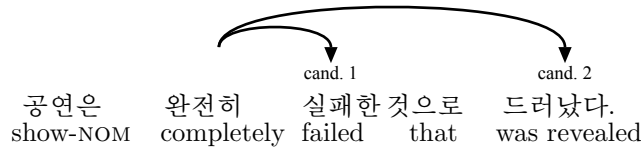


Figure 4.6: Two alternative head candidates of *완전히* (*wanjeonhi*; completely). The sentence interpreted as *It was revealed that the show was completely failed.* in English.

완전히 (*wanjeonhi*) in the sentence (4.15) and Figure 4.6 prefers the length 1 modification very much in such context.

$$P(\text{modification length} = 1 \mid \text{mag pv\text{g}-etm nbn-jca}) = 94.82$$

where “mag pv_g-etm nbn-jca” is the part-of-speech tag sequence³ of the word sequence “*완전히 실패한 것으로*” (*wanjeonhi silpaeha-n gut-euro*), which is the adverb and the context of it. Let’s see another example that shows the disambiguation power of the tendency for a word to have fixed modification distance in a certain context.

- (4.16) 하루를 자고나면 유럽의 지도가 바뀌는
haru-reul *jagonamyeon* *Europe-ui* *jido-ga* *bakkwineun* ...
 a day-ACC sleep Europe-GEN map-NOM changing
 ... that makes the map of Europe change in one night of sleep.

The correct head of *하루를* (*haru-reul*; a day-ACC; one night-ACC) is the *자고나면* (*jagonamyeon*; sleep) in the above sentence⁴. The head selection of the *하루를* (*haru-reul*) in this sentence can be done with the preference for the certain modification distance likewise. The probability of the length of the dependency relation from *하루를* (*haru-reul*) being 1 for the given context is

³The part-of-speech tag set in this dissertation is from the KAIST Treebank. The full list of the tag set is on Appendix.

⁴The full sentence is: *하루를 자고나면 유럽의 지도가 바뀌는 일대 격변이 이어나고 있는 것이다.* (*haru-reul jagonamyeon Europe-ui jido-ga bakkwineun ildae gyeokbyun-i ireonago inneun geosida.*), which means “There is a great upheaval making the map of Europe change in one night of sleep.”

$$P(\text{modification length} = 1 \mid \text{ncn-jco pvg-ecs nq-jcm ncn-jcs}) = 99.41$$

, which is nearly 100%. That means the noun ends with accusative case marker in the surrounding context almost, always modify the next following word. Of course, the head of the *하루를* (*haru-reul*) can be found with other methods as well. For instance, the likelihood of lexical co-occurrence between *하루를* (*haru-reul*) and its alternative head candidates can be used to select an appropriate head word. However, it uses lexical dependencies, which are usually sparse (Klein and Manning 2003).

4.4.2 Context for Word Dependency Prediction

(Lee 1997a) considers outer surface context of a constitute in estimating expanding probability of the constitute.

$$\begin{aligned} P(T, S) &\approx \prod_{rule \in T} P(rule \mid \text{outer surface context}) & (4.17) \\ &= \prod_{rule \in T} P(lhs \rightarrow rhs \mid \text{outer surface context}) \\ &= \prod_{rule \in T} P(rhs \mid lhs, \text{outer surface context}) \end{aligned}$$

By using the outer surface context, she can distinguish phrases that are different only in their coverages but equal in their phrase labels. She reports, by adding one part-of-speech tag of word for the left and the right context respectively, the labeled precision and recall increase by 5% and 7%.

We used similar context for our dependency parsing model. Namely, outside surface contexts of two depending words and the two words are used in estimating the dependency probability between two words. The dependency probability between two words can be more elaborately estimated when conditioned on the words that surround the dependency relation and the example of it was shown in the previous section. Here, we show another example of the ambiguity that can be helped by the outer context is shown in the following sentences.

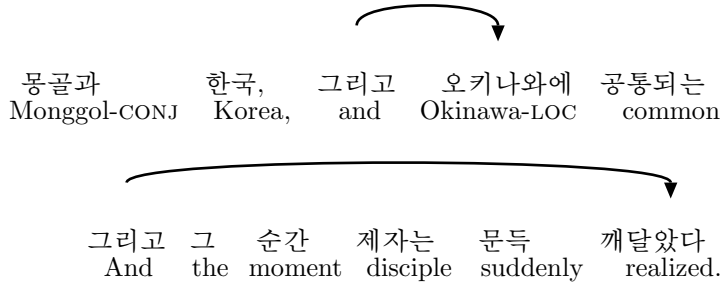


Figure 4.7: Dependency relations from *그리고* (*geurigo*) in the sentence (4.18) and (4.19)

(4.18) 몽골과 한국, 그리고 오키나와에 공통되는
Monggol-gwa Hanguk, geurigo Okinawa-e gongtongdoe-neun ...
 Monggol-CONJ Korea, and Okinawa-LOC common
 ... common in Monggol, Korean, and Okinawa

(4.19) 그리고 그 순간 제자는 문득 깨달았다.
geurigo geu sungan jeja-neun mundeuk kkaedara-tda.
 And the moment disciple-TOP suddenly realized.
 And, at the moment, the disciple suddenly realized.

When the conjunctive adverb *그리고* (*geurigo*; and) is used at the beginning of a sentence, it usually modifies the head of the whole sentence, which is the last verb of the sentence in many cases. However, if the conjunctive adverb appears next to a noun, it usually modifies another noun. That is, the left outer context helps disambiguating the head of the conjunctive adverb. So we consider the outer context in estimating the word dependencies.

4.5 Comparison with Other Models

In this section, we will show the difference of parameterization between the proposing model and other models. In many previous researches (Kim 1994; Collins 1996; Kim and Seo 1997; Kim et al. 1997), distance measure was used in measuring lexical dependency. e.g.

$$P(\text{head}(w_i) = w_j | w_i w_j \text{ dist}(w_i, w_j)) \quad (4.20)$$

and $\text{dist}(w_1, w_2)$ is a distance between two depending words. With the parsing model that measures the likelihood of a dependency like Equation (4.20), sparse data problem may arise. Consider the following three Korean sentences.

(4.21) 그가 바나나를 먹었다.
geu-ga banana-reul meogeotda.
 he-NOM banana-ACC ate.
 He ate a banana.

(4.22) 바나나를 집에서 먹었다.
banana-reul jib-eseo meogeotda.
 banana-ACC home-LOC ate.
 (He) ate a banana at home.⁵

(4.23) 나는 바나나를 학교와 집에서 먹었다.
na-neun banana-reul hakgyo-wa jib-eseo meogeotda.
 I-TOP banana-ACC school-CONJ home-LOC ate.
 I ate a banana at school and home.

Assume the first two sentences are used to train the parsing model. Then the acquired statistics of the lexical dependency between two words 바나나를 (*banana-reul*; banana-ACC) and 먹었다 (*meogeotda* ; ate) are

$$P(\text{related}(\text{banana-reul}, \text{meogeotda}, 1)) = 1$$

$$P(\text{related}(\text{banana-reul}, \text{meogeotda}, 2)) = 1$$

However, these precious lexical dependency probabilities acquired from the test sentences cannot be used in analyzing the third sentence (4.23), because the distance between two words, 바나나를 (*banana-reul*) and 먹었다 (*meogeotda*), is 3. e.g.

⁵Subject ellipsis is common in Korean

$$P(\text{related}(\text{banana-reul}, \text{meogeotda}, 3)) = 0$$

That's why some works, such as Seo et al. (1999), insists the distance is not meaningful in parsing the variable-word-order languages like Russian, Korean and Japanese.

But, despite the flaw, the distance measure between words is empirically proved to have an effect on improving disambiguation performance. To solve the data sparseness problem, first, we split the lexical dependency prediction from the modification length prediction. Second, we assume that the modification distance (or the length of a dependency relation) depends only on the modifier, not on the modifyee. This makes the sparseness problem less serious. Third, the surrounding surface word is added to the context to increase the precision of the prediction. Part-of-speech tags are used to express the surrounding surface context to alleviate the data sparseness problem,

Chapter 5

Statistical Dependency Parsing Model for Korean

5.1 The Statistical Parsing Model

We propose a new probabilistic parsing model for Korean. The model consists of two probabilities, which are the lexical dependency probability and the modification distance probability. We use a lexical dependency probability to reflect the selectional preferences between two words, and modification distance probability to reflect the modifying distance preference of a word. The former is used to predict the word dependency, while the latter predicts the modification distance of the modifier. First, we will introduce the lexical dependency probability and then we will explain the modification distance probability.

5.1.1 Lexical Dependency Probability

The lexical dependency probability between the two given words w_i and w_j is represented as follows:

$$P(\text{head}(w_i) = w_j | w_i, w_j) \tag{5.1}$$

$head(x)$ is the head word of the word x . This is the conditional probability of existing dependency relation for given two words w_i and w_j in a sentence. A word w is divided into content morpheme cm and a functional morpheme fm to reflect morphological characteristics of Korean.

$$P(head(w_i) = w_j | w_i w_j) = P(head(w_i) = w_j | cm_i fm_i cm_j fm_j) \quad (5.2)$$

For example, the lexical dependency probability between two words 밥을 (*bap-eul* ; meal-ACC) and 먹다 (*meog-da*; eat-END) is

$$\begin{aligned} P(head(w_i) = w_j | w_i w_j) &= P(head(bap-eul) = meog-da | bap-eul meog-da) \\ &= P(head(bap-eul) = meog-da | bap eul meog da) \end{aligned}$$

Because each word w_x has its part-of-speech tag t_x , the expression (5.1) can be rewritten as follows

$$P(head(w_i) = w_j | w_i w_j) = P(head(w_i) = w_j | w_i w_j t_i t_j) \quad (5.3)$$

The advantage of this probability is that it spontaneously reflects selectional preferences of lexical words. The selectional preference is very useful especially for analyzing variable word order languages like Korean, because these kind of languages have little constraint on word order. Therefore, the decisions during the parsing are made solely, or mostly, based on the selectional preference between words. And if the conditional part of Equation (5.3) is *backed-off* like

$$P(head(w_i) = w_j | w_i w_j t_i t_j) \approx P(head(w_i) = w_j | t_i t_j) \quad (5.4)$$

, the model can seamlessly consider context-free probabilistic dependency grammar, which usually defines the likelihood of part-of-speeches of two words having dependency relation to each other.

The lexical dependency probabilities measured from a Treebank were shown in Table 5.1. Since they are estimated with deleted interpolation for smoothing effect, the probabilities may be higher than we expect. For example, the lexical dependency likelihood of two words 책을 (*chaek-eul*; book-ACC) and 마시 (*masi*; drink) is over 50%. It means that if the two

	밥을 (meal-ACC)	책을 (book-ACC)	공기를 (air-ACC)
먹 (eat)	0.8599	0.4462	0.4511
마시 (drink)	0.4969	0.5032	0.5081
보 (see)	0.4578	0.5962	0.1172

Table 5.1: Lexical dependency probabilities between two words, measured from 30,000 sentences of a Treebank. In estimating the probabilities, deleted interpolation is used to alleviate lexical data sparseness

words appear together in one sentence, the probability of the two words depending each other is one in two. According to our linguistic knowledge, this doesn't make sense, because the expression 'drinking a book' is odd. But, as mentioned earlier, the probabilities are measured with smoothing techniques. The probabilities also contains the interdependency likelihood of the part-of-speech tags of those words, which are `ncn-jcm` and `pvg`. This is why the lexical dependency probabilities between *책을* (*chaek-eul*; book-ACC) and *마시* (*masi*; drink) are higher than our expectation. However, likely word pairs such as

<밥을 (*bap-eul*; meal-ACC), 먹 (*meog*; eat) >,
 <책을 (*chaek-eul*; book-ACC), 보 (*bo*; see) >, and
 <공기를 (*gonggi-reul*; air-ACC), 마시 (*masi*; drink) >

get higher dependency probabilities than other pairs do.

Table 5.2 shows that the lexical dependency probability not only reflects selectional preference, but also dependency rule probability. The first and second rows show the dependency probabilities between two words whose part-of-speech tags are `ncn-jcm`¹ and `pvg`. The nouns end with adnominal case particle cannot modify verbs in Korean syntax; The dependency relation between them is ungrammatical. Reflecting the grammar, all of the probabilities are 0. Meanwhile, the lower rows of the table shows the dependency probabilities between two

¹The part-of-speech `jcm` (adnominal case particle) is assigned to the genitive case marker *의* (*ui*).

	사람의 (person-GEN)	책의 (book-GEN)	나라의 (nation-GEN)
보 (see)	0.0000	0.0000	0.0000
먹 (eat)	0.0000	0.0000	0.0000
이름 (name)	0.4880	0.3872	0.3792
나이 (age)	0.3172	0.3120	0.3040

Table 5.2: Lexical dependency probabilities between two words that show the probabilities reflect the likelihood of grammar rules usage. The upper rows shows the dependency probability between two words whose part-of-speech tags are `ncn-jcm` and `pvg`; the two lower rows show probabilities between words whose part-of-speech tags are `ncn-jcm` and `ncn`

words whose part-of-speech tags are `ncn-jco` and `pvg`. The dependency relation between those words are grammatical allowable and, therefore, they have certain probabilities. Thus the lexical dependency probability is a measure of not only selectional preference, but also dependency rule likelihood.

Additional, outer contextual information is also considered in estimating the lexical dependency probability. As we discussed earlier, the part-of-speech tag of the word at the left of the modifier (t_{ol}) and the part-of-speech tag of the word at the right of the head word (t_{or}) are used to consider contextual information, when estimating the lexical dependency probability between the modifier and the head.

$$P(\text{head}(w_i) = w_j \mid w_i w_j t_{ol} t_{or}) = P(\text{head}(w_i) = w_j \mid w_i w_j ft_{i-1} ct_{j+1}) \quad (5.5)$$

where ct_n and ft_n are part-of-speech tag of the content morpheme and functional morpheme of the w_n , respectively.

5.1.2 Modification Distance Probability

A modification distance probability is defined to reflect our observation that the length of a modification, or dependency relation can be predicted with a modifier and its surrounding

context. This probability reflects the following two preferences:

1. Whether a modifier prefers long distance modification or local (short distance) modification.
2. If a modifier prefers local modification, which word in the local context is preferred as its head.

The modification distance probability is a conditional probability function for the random variable $length(w_i)$ given a modifier w_i and its local context Φ_i . The random variable $length(w_i)$ is :

$$length(w_i) = \Psi(d)$$

$$\Psi(d) = \begin{cases} d & \text{if } d < K \\ long & \text{else.} \end{cases}$$

where d is a surface length of a dependency relation between w_i and w_j ;

$$d = j - i$$

A constant K is the yardstick to decide whether a dependency relation is short or long. If a length of the dependency relation is longer than the constant K , then the dependency relation is considered to be *long*.

The modification distance probability of a certain $length(w_i)$ value for the given dependent word w_i and its surrounding context Φ_i is as follows:

$$P(length(w_i) | w_i \Phi_i) = P(length(w_i) | w_i t_{i-m} \cdots t_{i-1} t_{i+1} \cdots t_{i+n}) \quad (5.6)$$

, when the contextual information Φ_i is expressed with part-of-speech tag sequence. The size of context is determined by the constants m and n , which are empirically set. For example, the probability of the length 3 dependency relation starting from the word *완전히* (*wanjeonhi*) in the sentence (4.15), when m and n are 1 and 2, is

$$P(length(w_2) = \Psi(3) | w_2 \Phi_2) = P(length(wanjeonhi) = \Psi(3) | wanjeonhi \Phi_2)$$

$$\begin{aligned}
&= P(\text{length}(\text{wanjeonhi}) = \Psi(3) | \text{wanjeonhi } t_1 t_3 t_4) \\
&= P(\text{length}(\text{wanjeonhi}) = \Psi(3) | \text{wanjeonhi} \\
&\quad \text{ncn-jxt pvg-etm nbn-jca})
\end{aligned}$$

5.1.3 The Parsing Model

The most probable dependency tree T_{best} for a sentence S is the tree T that maximizes the conditional probability $P(T|S)$.

$$T_{best} = \operatorname{argmax}_T P(T|S) \quad (5.7)$$

Since T consists of a set of dependency relations, the probability of each tree T for a sentence S is estimated by a probability product of all dependency relations in the tree (Assuming independency between dependency relations). Formally,

$$\begin{aligned}
P(T|S) &= P(\text{dep}_1 \text{dep}_2 \cdots \text{dep}_{|S|-2} \text{dep}_{|S|-1} | S) \\
&= P(\text{dep}_1 | S) \cdot P(\text{dep}_2 | \text{dep}_1, S) \cdots P(\text{dep}_{|S|-1} | \text{dep}_1 \cdots \text{dep}_{|S|-2}, S) \\
&\approx \prod_{0 < i < |S|} P(\text{dep}_i | S) \quad (5.8)
\end{aligned}$$

We assume the dependency relation dep_i depends on the two words which the dependency relation links and surrounding context of the words. So,

$$\begin{aligned}
P(T|S) &= P(\text{dep}_1 \text{dep}_2 \cdots \text{dep}_{|S|-2} \text{dep}_{|S|-1} | S) \\
&\approx \prod_{0 < i < |S|} P(\text{dep}_i | S) \\
&\approx \prod_{0 < i < |S|} P(\text{dep}_i | w_i w_{h(i)} \Phi_i \Phi_{h(i)}) \quad (5.9)
\end{aligned}$$

when $h(x)$ is the position of the head of w_x and Φ_x a context around w_x . And a dependency relation dep_i defines a head of a word, and a length of the modification between them. Formally,

$$dep_i = \begin{bmatrix} head(w_i) & w_{h(i)} \\ length(w_i) & \Psi(h(i) - i) \end{bmatrix}$$

Then the conditional probability for the dep_i in Equation (5.9) becomes

$$P(dep_i | w_i w_{h(i)} \Phi_i \Phi_{h(i)}) \quad (5.10)$$

$$\begin{aligned} &= P(head(w_i) = w_{h(i)}, length(w_i) = \Psi(h(i) - i) | w_i w_{h(i)} \Phi_i \Phi_{h(i)}) \\ &= P(head(w_i) = w_{h(i)} | w_i w_{h(i)} \Phi_i \Phi_{h(i)}) \\ &\quad \cdot P(length(w_i) = \Psi(h(i) - i) | head(w_i) = w_{h(i)} w_i w_{h(i)} \Phi_i \Phi_{h(i)}) \quad (5.11) \end{aligned}$$

$$\begin{aligned} &\approx P(head(w_i) = w_{h(i)} | w_i w_{h(i)} \Phi_i \Phi_{h(i)}) \\ &\quad \cdot P(length(w_i) = \Psi(h(i) - i) | w_i \Phi_i) \quad (5.12) \end{aligned}$$

$$\begin{aligned} &\approx P(head(w_i) = w_{h(i)} | w_i w_{h(i)} ft_{i-1} ct_{h(i)+1}) \\ &\quad \cdot P(length(w_i) = \Psi(h(i) - i) | w_i \Phi_i) \quad (5.13) \end{aligned}$$

by a chain rule (5.11) and our assumption that the length of a dependency relation only depends on the modifier (5.12). By considering only the part-of-speech tag of the morpheme at right before the modifier and right next to the head in estimating lexical dependency probability, Φ_i and $\Phi_{h(i)}$ in Equation (5.12) becomes (5.13).

Therefore, the probability of the parse tree T for the sentence S becomes the product of Equation (5.10) :

$$\begin{aligned} P(T|S) &\approx \prod_{0 < i < |S|} P(dep_i | S) \quad (5.14) \\ &\approx \prod_{0 < i < |S|} P(head(w_i) = w_{h(i)}, length(w_i) = \Psi(h(i) - i) | w_i w_{h(i)} \Phi_i \Phi_{h(i)}) \\ &\approx \prod_{0 < i < |S|} P(head(w_i) = w_{h(i)} | w_i w_{h(i)} ct_{i-1} ft_{h(i)+1}) \\ &\quad \cdot P(length(w_i) = \Psi(h(i) - i) | w_i \Phi_i) \end{aligned}$$

As you see, the probability of a dependency tree becomes the product of the probability

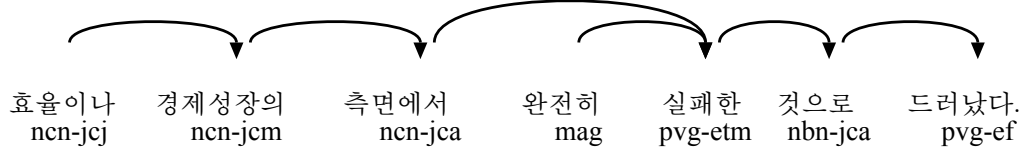


Figure 5.1: A dependency tree, which is part-of-speech tagged

of lexical dependency and the probability of length of modification. For example, Figure 5.1 shows a dependency tree for the following sentence.

(5.15)	효율이나	경제성장의	측면에서	완전히
	<i>hyoyur-ina</i>	<i>gyeongjeseongjang-ui</i>	<i>cheungmyeon-eseo</i>	<i>wanjeonhi</i>
	Efficiency-or	economic growth-GEN	aspect-LOC	completely
	실패한	것으로	드러났다.	
	<i>silpaeha-n</i>	<i>geos-euro</i>	<i>deureonatda.</i>	
	failed	that	was revealed	

It was revealed as complete failure in the aspect of efficiency or economic growth.

The probability of the dependency tree² is

$$\begin{aligned}
P(T|S) &= \prod_{0 < i < S} P(dep_i|S) \\
&\approx \prod_{0 < i < |S|} P(head(w_i) = w_{h(i)} | w_i w_{h(i)} ct_{i-1} ft_{h(i)+1}) \\
&\quad \cdot P(length(w_i) = \Psi(h(i) - i) | w_i \Phi_i) \\
&\approx \prod_{0 < i < |S|} P(head(w_i) = w_{h(i)} | w_i w_{h(i)} ct_{i-1} ft_{h(i)+1}) \\
&\quad \cdot P(length(w_i) = \Psi(h(i) - i) | w_i t_{i+1} t_{i+2}) \\
&= P(head(w_1) = w_2 | w_1 w_2 \langle s \rangle \mathbf{ncn}) \\
&\quad \cdot P(length(w_1) = \Psi(1) | w_1 \mathbf{ncn-jcj} \mathbf{ncn-jcm} \mathbf{ncn-jca}) \times
\end{aligned}$$

² m and n values for deciding local contextual pattern Φ_i are set to 0 and 3 in this example.

$$\begin{aligned}
& P(\text{head}(w_2) = w_3 \mid w_2 \ w_3 \ \text{j c j} \ \text{mag}) \\
& \quad \cdot P(\text{length}(w_2) = \Psi(1) \mid w_2 \ \text{ncn-jcm} \ \text{ncn-jca} \ \text{mag}) \times \\
& P(\text{head}(w_3) = w_5 \mid w_3 \ w_5 \ \text{jcm} \ \text{nbm}) \\
& \quad \cdot P(\text{length}(w_3) = \Psi(2) \mid w_3 \ \text{ncn-jca} \ \text{mag} \ \text{pvg-etm}) \times \\
& P(\text{head}(w_4) = w_5 \mid w_4 \ w_5 \ \text{jca} \ \text{nbm}) \\
& \quad \cdot P(\text{length}(w_4) = \Psi(1) \mid w_4 \ \text{pvg-etm} \ \text{ncn-jca} \ \text{pvg-ef}) \times \\
& P(\text{head}(w_5) = w_6 \mid w_6 \ w_6 \ \text{mag} \ \text{pvg}) \\
& \quad \cdot P(\text{length}(w_5) = \Psi(1) \mid w_5 \ \text{nbm-jca} \ \text{pvg-ef} \ \langle \mathbf{e} \rangle) \times \\
& P(\text{head}(w_6) = w_7 \mid w_6 \ w_7 \ \text{etm} \ \langle \mathbf{e} \rangle) \\
& \quad \cdot P(\text{length}(w_6) = \Psi(1) \mid w_6 \ \text{pvg-ef} \ \langle \mathbf{e} \rangle \ \langle \mathbf{e1} \rangle)
\end{aligned}$$

, where $w_1 = \text{효율이나}$ (*hyoyur-ina*), $w_2 = \text{경제성장의}$ (*gyeongjeseongjang-ui*), $w_3 = \text{측면에서}$ (*cheungmyeon-eseo*), $w_4 = \text{완전히}$ (*wanjeonhi*), $w_5 = \text{실패한}$ (*silpaeha-n*), $w_6 = \text{것으로}$ (*geos-euro*), and $w_7 = \text{드러났다}$ (*deureonatda*).

5.2 The Statistical Parser & Parameter Estimation

5.2.1 The Parser

A parser using the proposed parsing model is designed. The parser selects a tree T_{best} among all parses for a sentence S

$$T_{best} = \underset{T}{\operatorname{argmax}} P(T|S) \quad (5.16)$$

And the parser uses some simple constraints to reflect the syntactic characteristic of Korean, to prevent over-generation of dependency arcs, and to promote efficiency and accuracy. The constraints are

- Every modifier precedes its head.
- Every dependency relation don't cross.

- Every verb isn't allowed to have more than one subject and object.

The parser uses a simple bottom-up chart parsing algorithm. It doesn't use any explicit grammar; any dependency relation with a part-of-speech tags of words which has been seen in training data will be considered valid.

5.2.2 Parameter Estimation

Parameters for the parsing model is estimated as following:

Lexical Dependency Probability

The parameters for the lexical dependency probability is measured as

$$\begin{aligned}
P(\text{head}(w_i) = w_j \mid w_i \ w_j \ ft_{i-1} \ ct_{j+1}) & \quad (5.17) \\
= P(\text{head}(w_i) = w_j \mid w_i \ w_j \ t_i \ t_j \ ft_{i-1} \ ct_{j+1}) \\
= P(\text{head}(w_i) = w_j \mid cm_i \ fm_i \ cm_j \ fm_j \ ct_i \ ft_j \ ct_i \ ft_j \ ft_{i-1} \ ct_{j+1}) \\
= F(\text{head}(w_i) = w_j \mid cm_i \ fm_i \ cm_j \ fm_j \ ct_i \ ft_j \ ct_i \ ft_j \ ft_{i-1} \ ct_{j+1})
\end{aligned}$$

when $F(\cdot)$ is the maximum likelihood estimation $P(\cdot)$.

$$\begin{aligned}
F(\text{head}(w_i) = w_j \mid cm_i \ fm_i \ cm_j \ fm_j \ ct_i \ ft_j \ ct_i \ ft_j \ ft_{i-1} \ ct_{j+1}) \\
= \frac{C(\text{head}(w_i) = w_j \ cm_i \ fm_i \ cm_j \ fm_j \ ct_i \ ft_j \ ct_i \ ft_j \ ft_{i-1} \ ct_{j+1})}{C(cm_i \ fm_i \ cm_j \ fm_j \ ct_i \ ft_j \ ct_i \ ft_j \ ft_{i-1} \ ct_{j+1})}
\end{aligned}$$

To cope with sparse data problem, linear interpolation is used to estimate the parameter (Figure 5.2).

Modification Distance Probability

Similarly, parameter for the modification distance probability

$$P(\text{length}(w_i) = \Psi(h(i) - i) \mid w_i \ \Phi_i) \quad (5.18)$$

$$\begin{aligned}
& P(\text{head}(w_i) = w_j \mid cm_i fm_i cm_j fm_j ct_i ft_i ct_j ft_j ft_{i-1} ct_{j+1}) \\
&= \frac{1}{2} \cdot P(\text{head}(w_i) = w_j \mid cm_i fm_i cm_j fm_j ct_i ft_i ct_j ft_j) \\
&\quad + \frac{1}{2} \cdot P(\text{head}(w_i) = w_j \mid cm_i cm_j ct_i ft_i ct_j ft_j ft_{i-1} ct_{j+1})
\end{aligned}$$

$$\begin{aligned}
& P(\text{head}(w_i) = w_j \mid cm_i fm_i cm_j fm_j ct_i ft_i ct_j ft_j) \\
&= \lambda \cdot F(\text{head}(w_i) = w_j \mid cm_i fm_i cm_j fm_j ct_i ft_i ct_j ft_j ft_{i-1} ct_{j+1}) \\
&\quad + (1 - \lambda) \cdot P(\text{head}(w_i) = w_j \mid cm_i cm_j ct_i ft_i ct_j)
\end{aligned}$$

$$\begin{aligned}
& P(\text{head}(w_i) = w_j \mid cm_i cm_j ct_i ft_i ct_j ft_j) \\
&= \lambda \cdot F(\text{head}(w_i) = w_j \mid cm_i cm_j ct_i ft_i ct_j) \\
&\quad + (1 - \lambda) \cdot \left\{ \frac{1}{2} P(\text{head}(w_i) = w_j \mid cm_j ct_i ft_i ct_j) \right. \\
&\quad \quad \left. + \frac{1}{2} P(\text{head}(w_i) = w_j \mid cm_i ct_i ft_i ct_j) \right\}
\end{aligned}$$

Interpolations for estimating the lexical dependency probability (continued)

$$\begin{aligned}
&= P(\text{length}(w_i) = \Psi(h(i) - i) \mid w_i t_{i-m} \cdots t_{i-1} t_{i+1} \cdots t_{i+n}) \\
&= P(\text{length}(w_i) = \Psi(h(i) - i) \mid w_i t_{i-m, i-1} t_{i+1, i+n})
\end{aligned}$$

is estimated with smoothing. Back-off smoothing is used to estimate the parameter (Figure 5.3).

$$\begin{aligned}
& P(\text{head}(w_i) = w_j \mid cm_j \ ct_i \ ft_i \ ct_j) \\
&= \lambda \cdot F(\text{head}(w_i) = w_j \mid cm_j \ ct_i \ ft_i \ ct_j) \\
&\quad + (1 - \lambda) \cdot P(\text{head}(w_i) = w_j \mid ct_i \ ft_i \ ct_j)
\end{aligned}$$

$$\begin{aligned}
& P(\text{head}(w_i) = w_j \mid cm_i \ ct_i \ ft_i \ ct_j) \\
&= \lambda \cdot F(\text{head}(w_i) = w_j \mid cm_i \ ct_i \ ft_i \ ct_j) \\
&\quad + (1 - \lambda) \cdot P(\text{head}(w_i) = w_j \mid ct_i \ ft_i \ ct_j)
\end{aligned}$$

$$\begin{aligned}
& P(\text{head}(w_i) = w_j \mid cm_i \ cm_j \ ct_i \ ft_i \ ct_j \ ft_{i-1} \ ct_{j+1}) \\
&= \lambda \cdot F(\text{head}(w_i) = w_j \mid cm_i \ cm_j \ ct_i \ ft_i \ ct_j \ ft_{i-1} \ ct_{j+1}) \\
&\quad + (1 - \lambda) \cdot P(\text{head}(w_i) = w_j \mid cm_j \ ct_i \ ft_i \ ct_j \ ft_{i-1} \ ct_{j+1})
\end{aligned}$$

$$\begin{aligned}
& P(\text{head}(w_i) = w_j \mid cm_j \ ct_i \ ft_i \ ct_j \ ft_{i-1} \ ct_{j+1}) \\
&= \lambda \cdot F(\text{head}(w_i) = w_j \mid cm_j \ ct_i \ ft_i \ ct_j \ ft_{i-1} \ ct_{j+1}) \\
&\quad + (1 - \lambda) \cdot P(\text{head}(w_i) = w_j \mid ct_i \ ft_j \ ct_j \ ft_j \ ft_{i-1} \ ct_{j+1})
\end{aligned}$$

$$\begin{aligned}
& P(\text{head}(w_i) = w_j \mid ct_i \ ft_j \ ct_j \ ft_{i-1} \ ct_{j+1}) \\
&= \lambda \cdot F(\text{head}(w_i) = w_j \mid ct_i \ ft_j \ ct_j \ ft_{i-1} \ ct_{j+1}) \\
&\quad + (1 - \lambda) \cdot P(\text{head}(w_i) = w_j \mid ct_i \ ft_i \ ct_j)
\end{aligned}$$

Figure 5.2: Interpolations for estimating the lexical dependency probability

if $C(w_i t_{i-m,i-1} t_{i+1,i+n}) > N$

$$P(e|w_i\Phi_i) = F(e | w_i t_{i-m,i-1} t_{i+1,i+n})$$

if $C(w_i t_{i-m+1,i-1} t_{i+1,i+n}) + C(w_i t_{i-m,i-1} t_{i+1,i+n-1}) + C(t_i t_{i-m,i-1} t_{i+1,i+n}) > N$

$$P(e|w_i\Phi_i) = \frac{C(e|w_i t_{i-m+1,i-1} t_{i+1,i+n}) + C(e|w_i t_{i-m,i-1} t_{i+1,i+n-1})}{C(w_i t_{i-m+1,i-1} t_{i+1,i+n}) + C(w_i t_{i-m,i-1} t_{i+1,i+n-1})} + \frac{C(e|t_i t_{i-m,i-1} t_{i+1,i+n})}{C(t_i t_{i-m,i-1} t_{i+1,i+n})}$$

else $P(e|w_i\Phi_i) = F(e | t_i)$

Figure 5.3: Procedure of back-off smoothing for modification distance. e is the abbreviation of the event $length(w_i) = \Psi(h(i) - i)$

5.3 Experiments

5.3.1 Experimental Setup

All experiments are done with the dependency-tagged KAIST Treebank from KAIST Language Resources (Choi 2001). The Treebank consists of 31,080 sentences and stored in 100 files. We use the 91 files³ (27,694 sentences) as the training data and the 9 files⁴(3,386 sentences) as the heldout testing data. The average lengths of the sentences are 12.2 words and 12.45 words for training and testing set, respectively.

The input for the parser is part-of-speech tagged, and AUXP (AUXiliary Phrase) bracketed. Figure 5.4 shows some examples of sentences after AUXP chunking is applied. The AUXP bracketing is done with a simple rule-based AUXP chunker⁵. Instead of the part-of-speech tags, the tag AUXP is used as a part-of-speech information for the AUXP chunked words. For example, the part-of-speech tags of the following sentence

(5.19) *거기에 새로운 불을 붙여야 한다.*
geogi-e saero-un bul-eul budy-eoya ha-nda
there-LOC new fire-ACC set-OBL do-END

New fire has to be set there.

are “npd-jca paa-etm ncn-jco pvg-ecx px-ef-sf”. However, the morpheme sequence *어야 하* (-*eoya ha*) is bracketed as an AUXP, and , as the result, the input sentence becomes

(5.20) *거기에 새로운 불을 붙이-는다.*
geogi-e saero-un bul-eul budy-nda.
there-LOC new fire-ACC set-END

and their part-of-speech tags become “npd-jca paa-etm ncn-jco pvg-AUXP-ef-sf”. As you see in this example, AUXP chunking sometimes reduce the length of sentences. After AUXP chunking, the average length of sentences in the training and testing data decreases to 11.13 and 11.31 words.

³Files from m2ta_04 to mh2_0185

⁴Files from mh2_0190 to mh2_0372

⁵Details are on Appendix

*geu-ga eo-lin si-jeol-e jungbyeong-eul alh-**ass-eoss-da**.*
→ *geu-ga eo-lin si-jeol-e jungbyeong-eul alh-**AUXP-da**.*

He was seriously ill when he was a kid. (그가 어린 시절에 증병을 앓았다.)

*gae-hwa-pa-ui jungsim in-mul-eun KimOkGyun-i-**eoss-da**.*
→ *gae-hwa-pa-ui jungsim in-mul-eun KimOkGyun-i-**AUXP-da**.*

Kim, Ok-Gyun was at the center of the Civilizing faction. (개화파의 중심 인물은 김옥균이었다.)

*geu-neun bam-e jamja-**l su iss-do-log naj-e un-dongha-ess-da**.*
→ *geu-neun bam-e jamja-**AUXP-do-log naj-e un-dongha-AUXP-da**.*

He exercised during the day in the hope that he would be able to sleep at night. (그는 밤에 잠잘 수 있도록 낮에 운동하였다.)

*han beon mal-ha-e **bo-ass-eul ppun-i-da**.*
→ *han beon mal-ha-**AUXP-da**.*

I only said it. (한 번 말해 보았을 뿐이다.)

Figure 5.4: Examples of AUXP chunking

5.3.2 Experiment 1 : Context-Window Size for Modification Distance Decision

First of all, we evaluate our assumption, that is *length of a modification relation can be determined by a modifier and its local contextual pattern*. To do this, we made a classifier using the modification distance probability that models our assumption statistically. The classifier decide the length of a modification relation for the given modifier w_i and its context Φ_i .

$$\begin{aligned} \text{modification distance} &= \operatorname{argmax}_{d \in Dist} P(\text{length}(w_i) = d | w_i \Phi_i) \\ &= \operatorname{argmax}_{d \in Dist} P(\text{length}(w_i) = d | t_{i-n} \cdots w_i \cdots t_{i+m}) \end{aligned} \quad (5.21)$$

where $Dist$ is the range of the Ψ function.

$$Dist = \{1, 2, \cdots K - 1, Long\}$$

We experiment with changing n and m from 0 to 2, while k changes from 1 to 3. We used F_1 measure for evaluating the classifier.

$$\begin{aligned} \text{Precision} &= \frac{\text{number of instances correctly classified}}{\text{number of instances that the classifier classified}} \\ \text{Recall} &= \frac{\text{number of instances correctly classified}}{\text{number of all instances}} \\ F_1\text{-measure} &= \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \end{aligned}$$

The experimental result is on Table 5.3. It tells that considering wider context does not always induce more accurate classification. The best result is acquired when m and n are 0 and 2. This means the left context hardly affect the performance of deciding modification distances⁶.

⁶(Sekine, Uchimoto, and Isahara 2000) reported that similar characteristic is observed for Japanese too. Based on his experiment with humans, it is true more than 90% of the time for Japanese.

context size		<i>Dist</i>		
<i>m</i>	<i>n</i>	{1, long} (<i>k</i> =2)	{1,2, long} (<i>k</i> =3)	{1,2,3, long} (<i>k</i> =4)
0	0	0.797	0.757	0.729
1	0	0.883	0.760	0.728
2	0	0.795	0.755	0.725
0	1	0.883	0.830	0.793
1	1	0.896	0.836	0.799
2	1	0.849	0.795	0.763
0	2	0.927	0.879	0.839
1	2	0.895	0.851	0.814
2	2	0.816	0.775	0.747
0	3	0.872	0.831	0.800
1	3	0.831	0.793	0.770
2	3	0.795	0.755	0.731

Table 5.3: Experimental result (in F_1 -score) for the modification distance classifier, with various m (left context size) , n (right context size) , and k (class size) values

Right context size bigger than 3 does not help the classification too. The reason of this is the sparse data problem. Figure 5.5 shows an example of dependency relations starts from the word 동시에 (*dongsi-e* ; same time-TMP) and their modification distance probability in the sentence :

(5.22) 동시에 동서간의 냉전체제도 와해되고
dongsi-e *dongeogan-ui* *naengjeoncheje-do* *wahaedoego* ...
same time-TMP East and West-GEN cold war-AUX is collapsed

At the same time, the cold war between the East and the West is collapsed ...

The correct head of 동시에 (*dongsi-e*) is 와해되고 (*wahaedoego*). The probabilities at upper arcs of the figure are the modification distance probabilities when the window size of

the right context (n) is 2 while the lowers are 3. For both cases, estimating the probabilities with maximum likelihood estimation is impossible because of the data sparseness. So back-off smoothing is used. When n is 2, the right context size n is backed-off to 1 to compute the probability for the length x dependency relation from w_0 , *wanjeon-hi*

$$\begin{aligned}
 P(\text{length}(w_0) = \Psi(x) \mid w_0 \text{ ncn-jcm ncn-jxc}) & \quad (5.23) \\
 \approx \frac{C(\text{length}(w_0) = \Psi(x) \mid w_0 \text{ ncn-jcm})}{\sum_{y \in Dist} C(\text{length}(w_0) = y \mid w_0 \text{ ncn-jcm})}
 \end{aligned}$$

If n is 3, however, the back-off process changes, because the smoothing strategy used here only allows two phases of back-off. So, first, n is decreased to 2 to estimate the probability, and it fails. And then, the second-level (or the last-level) back-off phase, which use only part-of-speech tags of w_i to estimate the length of the dependency relation from w_i , and which is usually inaccurate, is applied:

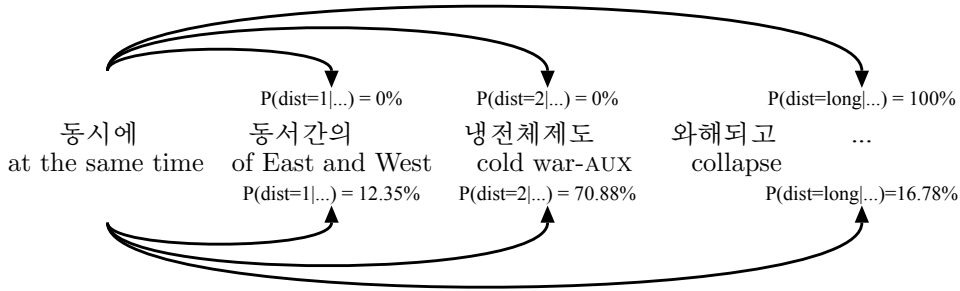
$$\begin{aligned}
 P(\text{length}(w_0) = \Psi(x) \mid w_0 \text{ ncn-jcm ncn-jxc paa-ecs}) & \quad (5.24) \\
 \approx \frac{C(\text{length}(w_0) = \Psi(x) \mid t_0)}{\sum_{y \in Dist} C(\text{length}(w_0) = y \mid t_0)} \\
 \approx \frac{C(\text{length}(w_0) = \Psi(x) \mid \text{ncn-jca})}{\sum_{y \in Dist} C(\text{length}(w_0) = y \mid \text{ncn-jca})}
 \end{aligned}$$

Using the last-level of the back-off, the classification with $n=3$ is inaccurate in the case of Figure 5.5. More sophisticated smoothing method may alleviate this problem and result better performance with bigger contextual window. But it may require higher computation and spatial complexity.

Meanwhile, the performance of the classifier increases as the value of K decreases. It is because a smaller K decreases the number of distance class, which is K , and classification becomes easier for smaller and more generic class. We selected the values for m , and n as 0 and 2 through this experiment, but could not decide the value for K . Although the performance of classifier with bigger K is worse, it might be more helpful for the parser to have probabilities for more subdivided distances.

Figure 5.6 is an example that shows the effect of different K makes. The upper arcs show modification distance probability when K is 2. The lower arcs show the probability

when $m=0, n=2$



when $m=0, n=3$

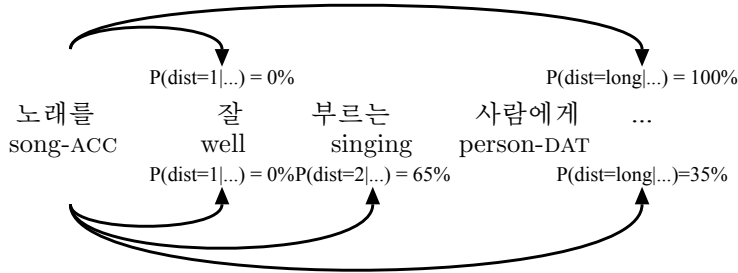
Figure 5.5: Effect of different right contextual size. The classifier made an error when $n=3$.

when K is 3. When K is 2, the only information we can get from the modification distance probability is that the modifier *norae-reul* (song-ACC) does not modify the next word *jal* (well) . However, this independency can be known by simple dependency rule probability because an object noun never modifies an adverb. So the modification distance probability is not helpful when K is 2. However, when the value of K is 3, the modification distance probability assigns higher probability for length 2 modification relation, which cannot be considered with the simple dependency probability. So we will not determine the value K here, but use all K for the following experiments.

5.3.3 Experiment 2 : Parsing Performance

Here, the performance of the parser that uses the proposed statistical parsing model is evaluated. An arc-based precision and recall measure, that are similar to PARSEVAL measures (Black, Abney, Flickenger, Gdaniec, Grishman, Harrison, Hindle, Ingria, Jelinek, Klavans, Liberman, Marcus, Roukos, Santorini, and Strzalkowski 1991) are used to evaluate the parser :

when $K = 2 : Dist = \{1, long\}$



when $K = 3 : Dist = \{1, 2, long\}$

Figure 5.6: Effect of different K makes. Comparison of the modification distance probability from the word *norae-reul*, when $k = 2$ and $k = 3$. As K gets bigger, the modification distance probability may be more helpful.

$$\text{Arc-based Precision (AP)} = \frac{\text{number of correct arcs in proposed parse}}{\text{number of arcs in proposed parse}}$$

$$\text{Arc-based Recall (AR)} = \frac{\text{number of correct arcs in proposed parse}}{\text{number of arcs in gold-standard parse}}$$

$$\text{Arc-based } F_1\text{-measure} = \frac{2 \cdot AP \cdot AR}{AP + AR}$$

Exact-matching rate is also used :

$$\text{Exact-Matching} = \frac{\text{number of correctly analyzed sentences}}{\text{number of sentences in test data}}$$

Table 5.4 shows the results for for K is 2, 3 and 4. The interesting point here is that the parsing performance for testing data are similar for any k values. It shows that even though the larger K shows worse classification performance in the previous experiment, the probabilities assigned to subdivided class due to larger K helps disambiguation.

	Measures	$K=2$ ({1, long})	$K=3$ ({1,2, long})	$K=4$ ({1,2,3, long})
Training Set	Arc F_1	0.9897	0.9842	0.9804
	ExactMatch	0.9023	0.8631	0.8333
Testing Set	Arc F_1	0.8581	0.8668	0.8674
	ExactMatch	0.3452	0.3467	0.3405

Table 5.4: Parsing Results on the training and testing data

5.3.4 Experiment 3 : Comparison with Other Parsers

In order to compare the proposed model with the other approaches, we have implemented four more parsers that use different models for syntactic disambiguation. All of them consider the distance measure somehow.

Model1 Lexical dependency probability model (the baseline)

$$P(dep_i|S) \approx P(head(w_i) = w_j | w_i w_j)$$

Model2 Lexical dependency probability model that distinguishes dependencies between adjacent words from other lexical dependencies (similar to the language model of Kim (1994), but lexicalized).

$$P(dep_i|S) = P(head(w_i) = w_j | w_i w_j d)$$

$$where \quad d = \begin{cases} 1 & \text{if } j - i = 1 \\ 2 & \text{if } j - i > 1 \end{cases}$$

Model3 Lexical dependency probability model combined with triplet/quadruplet head candidate decision model ⁷ (similar to Kanayama, Torisawa, Mitsuichi, and Tsujii (1999)’s model but consider lexical dependency likelihood in addition).

⁷Kanayama, Torisawa, Mitsuichi, and Tsujii (1999)’s model requires hand-crafted grammar, such as

$$P(dep_i|S) \approx P(head(w_i) = w_j | w_i w_j) \times P(head(w_i) = w_j | cd(i, 1) cd(i, 2) cd(i, 3))$$

, where

$$P(head(w_i) = w_j | cd(i, 1) \dots) = \begin{cases} 1 & \text{if } w_j = cd(i, 1) \ \& \ noc(i) = 1 \\ \epsilon & \text{if } w_j \notin \{cd(i, 1), cd(i, 2), cd(i, last)\} \\ P(k | w_i, cd(i, 1), cd(i, last)) & \\ \quad \text{if } noc(i) = 2 \ \& \ w_j \in \{cd(i, 1), cd(i, last)\} \\ P(k | w_i, cd(i, 1), cd(i, 2), cd(i, last)) & \\ \quad \text{if } noc(i) \geq 3 \ \& \ w_j \in \{cd(i, 1), cd(i, 2), cd(i, last)\} \end{cases}$$

$cd(i, y)$ is the part-of-speech tag for the y th head candidate of w_i and noc is the number of head candidate for w_i .

Model4 Parsing model based on bigram lexical dependency and distance between depending words (similar to parsing model of Collins (1996) and Kim and Seo (1997)).

$$P(dep_i) = P(head(w_i) = w_j | w_i w_j \Delta)$$

, where $\Delta = h - i$

All of these models are trained and tested on the same data that was used for the proposed model and the results are shown in Table 5.5.

Our model does not perform well in the training set, compared to other models. **Model2** and **Model4** achieve almost 100% arc F -score for the training set. It is because they are highly lexicalized models; Those models memorize the lexical dependency relation with the distance between them. **Model3** shows a worse result than our model for the training set. The triplet/quadruplet model used in **Model3** assumes that a head of a word is one among the nearest, second nearest, or the last head candidates from the dependent word. However, according to our survey on the training data, only 91.48% of heads are among the three HPSG. Instead of HPSG, we used *Treebank grammar* (Charniak 1996) for selecting head candidates. i.e. a set of dependency rules whose frequency is more than 1 in the training corpus is used as grammar.

	Measures	Model1	Model2	Model3	Model4	Proposed
Training Set	Arc F_1 score	0.9937	0.9953	0.9443	0.9962	0.9804
	Exact Matching	0.9476	0.9600	0.6444	0.9665	0.8336
Testing Set	Arc F_1 score	0.7846	0.8302	0.8321	0.8370	0.8674
	Exact Matching	0.2451	0.2548	0.2826	0.2569	0.3405

Table 5.5: Parsing Result for various parsers ($K = 4$ for **Proposed** model)

head candidates for the training data and this means that the triplet/quadruplet model is useful for only 91.48% of words⁸. This makes the performance of **Model3** poor.

Let’s see the results on the testing set. All models using the distance measure (**Model1** \sim **Proposed**) performs better than the baseline **Model1**. This shows the distance measure is still useful in analyzing variable word order languages. The proposed model outperformed all other models in the experiment, for parsing the heldout data.

The result of 10-fold cross validation to the KAIST Treebank for **Model4** and **Proposed** model is Table 5.6. The proposed model does better than the **Model4** in the cross validation as well. And the improvement (+2.4% Arc-based F score from Model4) is statistically meaningful.

5.4 Result Analysis

Table 5.7 shows the effect of additional information in parsing performance. Considering outer context (two part-of-speech tags surrounding a dependency relation) in measuring lexical dependency probability increases 1.4% arc-based F_1 score only. Using the modification distance probability is much more successful; it increases the F_1 score 7.5 % higher from the baseline, and 7.0 % higher from the performance of lexical dependency parsing model that

⁸To avoid parsing failure, we give very small probability to events, even though their likelihood are 0. That’s why the Arc F_1 score of **Model3** on the training data is higher than 91.48%.

set	Model4		Proposed	
	Arc F_1	EM	Arc F_1	EM
0	0.8355	0.2677	0.8571	0.3314
1	0.8354	0.2677	0.8587	0.3324
2	0.8361	0.2796	0.8594	0.3459
3	0.8340	0.2712	0.8573	0.3349
4	0.8363	0.2693	0.8589	0.3362
5	0.8365	0.2719	0.8597	0.3430
6	0.8364	0.2841	0.8573	0.3459
7	0.8341	0.2793	0.8587	0.3375
8	0.8308	0.2738	0.8580	0.3517
9	0.8336	0.2600	0.8600	0.3443
Avg.	0.8349	0.2725	0.8585	0.3403

Table 5.6: The result of 10-fold cross validation to the KAIST Treebank for **Model4** and **Proposed** model

Measures		LexDep	LexDep	LexDep	LexDep
		(BaseLine)	+OutCntxt	+ ModDist	+ Both
Testing Set	Arc F_1 score	0.7846	0.7983	0.8594	0.8675
	Exact Matching	0.2451	0.2628	0.3195	0.3405

Table 5.7: Effect of additional information for the testing set. LexDep, OutCntxt and ModDist stand for lexical dependency probability (not considering outer context), outer context, and modification distance probability. Both means OutCntxt+ModDist.

considers outer context. From the result, we can know the modification distance probability is a core factor that increases the parsing performance. Without considering surrounding context in estimating the lexical dependency probability, our model performs better than **Model2**, **Model3** and **Model4**, which are using distance measures in other ways (Table 5.7).

To investigate the exact effect from each additional context, we investigate the parsers’ performance for every length of dependency relations. Figure 5.7 shows the per-length performance of Model3, Model4 and the proposed model. It shows the proposed model outperforms other models in finding dependency relations of all lengths. This is the effect of the outer contextual information and modification distance probability.

However, removing the outer contextual information from our model causes different result. Figure 5.8 shows the per-length performance of the proposed model without the outer context. It performs well for finding the short-distance head, compared to other models. But it performs worse than Model3 in finding the long-distance heads. Model3’s outstanding performance in deciding long-distance head comes from consideration of the last head candidate. Recall Model3, the triplet/quadruplet model. It assumes that a head of a word is one among the nearest, the second nearest, and the last head candidates from the dependent word. That’s why Model3 performs as well as our model (without contextual information) for deciding short dependencies, and much better for long dependencies, but

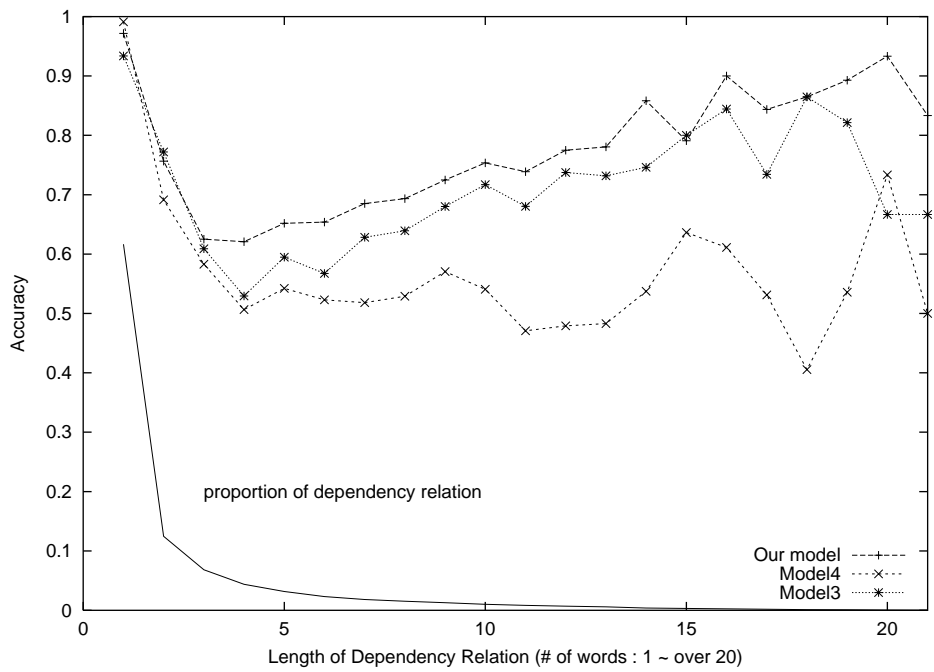


Figure 5.7: Arc-based accuracy vs. length of dependency relations figures for the proposed model, Model3 and Model4 in the test data

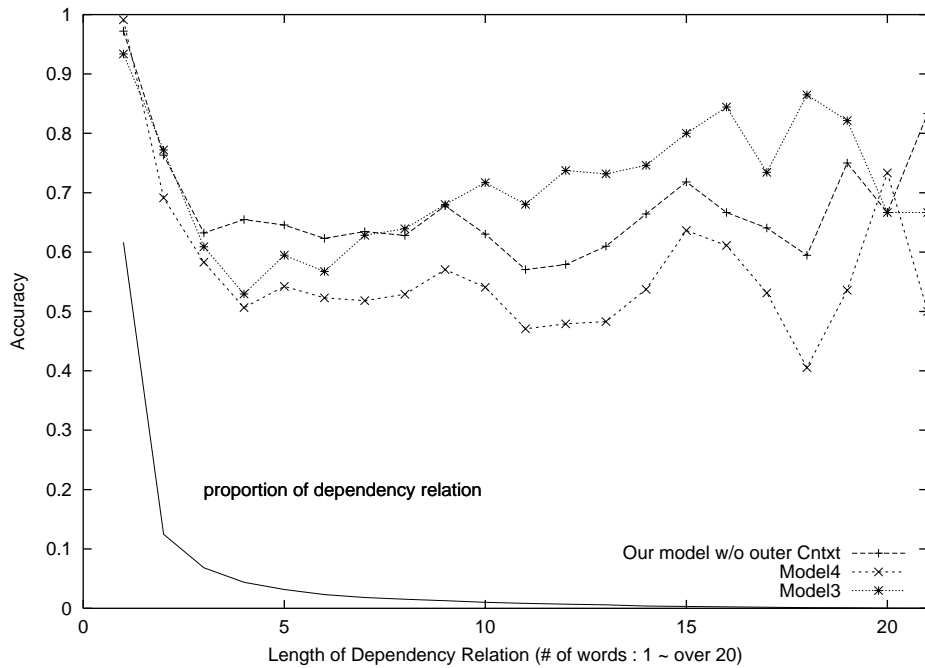


Figure 5.8: Arc-based accuracy vs. length of dependency relations figures for the proposed model without outer context, Model3 and Model4 in the test data

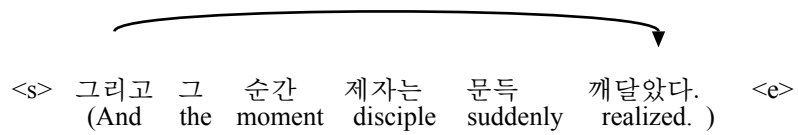


Figure 5.9: Effect of outer context in estimating lexical dependency. <s> and <e> are sentence start/end marker

Chapter 6

Discussion

6.1 Unlexicalized Parsing

6.1.1 Motivation

Lexical information has been widely used to achieve a high degree of parsing accuracy. Parsers with lexicalized language models (Collins 1999; Charniak 1999; Charniak 2001) have shown the state-of-the-art performances in analyzing English. Most of parsers developed recently use lexical features for syntactic disambiguation, whether they use a phrase structure grammar or a dependency grammar, regardless of languages they deal with.

However, some recent researches have found out that the lexicalization does not play a big role in parsing with probabilistic context-free grammars or PCFG. Gildea (2001) shows that the lexical bigram information does not contribute to the performance improvement of a parser. He says the lexical bigram statistics appear to be corpus-specific and they are no use when attempting to generalize to new training data. According to his experiment, removing the lexicalized statistics from Model 1 of Collins (1997) reduces parsing performance by less than 0.5% when testing the parser from the same domain as the training data. The unlexicalization doesn't affect the performance at all for test data from a different domain. Klein and Manning (2003) concludes that the fundamental sparseness of the lexical dependency information from parsed training corpora causes this result. For example, given by

(Klein and Manning 2003), many very plausible word pairs occur only once, such as *stocks stabilized*, while many others occur not at all, for example *stock skyrocketed*, in Penn Wall Street Journal corpora. Based on Gildea (2001)’s experiment, we can conclude the high performances of their parsers are not the effect of the lexical dependency information, but the contribution of the model itself. Those researchers parameterize the parsing process and build a mathematical model carefully to achieve high accuracy in analyzing sentence structure. Of course, the parameterization includes the lexicalization, but we think the prime factor of the performance improvement is the parameterization itself, not the lexicalization.

These reports motivate us to analyze the performance of our parsing model, which is lexicalized. This chapter investigates the effect of the unlexicalization in a dependency parser for Korean. The result shown above is the story of analyzing fixed word order languages, e.g. English, with a phrase structure grammar. We are anxious that an unlexicalized dependency parser for variable word order language can achieve high accuracy as the unlexicalized PCFG parser for English does. And then we suggest some techniques to achieve higher accuracy with unlexicalized parsers.

6.1.2 Investigation on Lexical Bigram Distribution

As stated in the previous chapter, bigram lexical dependency information has been considered to reflect selectional preference between words. By considering the selectional preference of words, a parser can select the appropriate parse easily among all possible parse structures. In fact, parsers using lexical dependency probability has performed very well (Magerman 1995; Collins 1996; Collins 1997; Charniak 2001). However, there is a simple, but big problem for the lexical dependency; the sparseness. The probability of lexical dependency between words can hardly be estimated if the lexical words co-occur sparsely. We investigate the distribution of lexical word pair¹ that we use in our parsing model to check how much sparse they are. The distribution is based on the frequencies of lexical word pairs

¹We actually investigate the frequency of *morpheme ternary* instead of *word pair*. Since a Korean word contains a content morpheme and functional morpheme, a word pair becomes a set of four morphemes. This may cause lexical sparseness worse. So we use the morpheme ternary $\langle cm_i, fm_i, cm_j \rangle$ to represent the word pair $\langle w_i w_j \rangle$. Using the ternary is sufficient to reflect the selectional preference between two words.

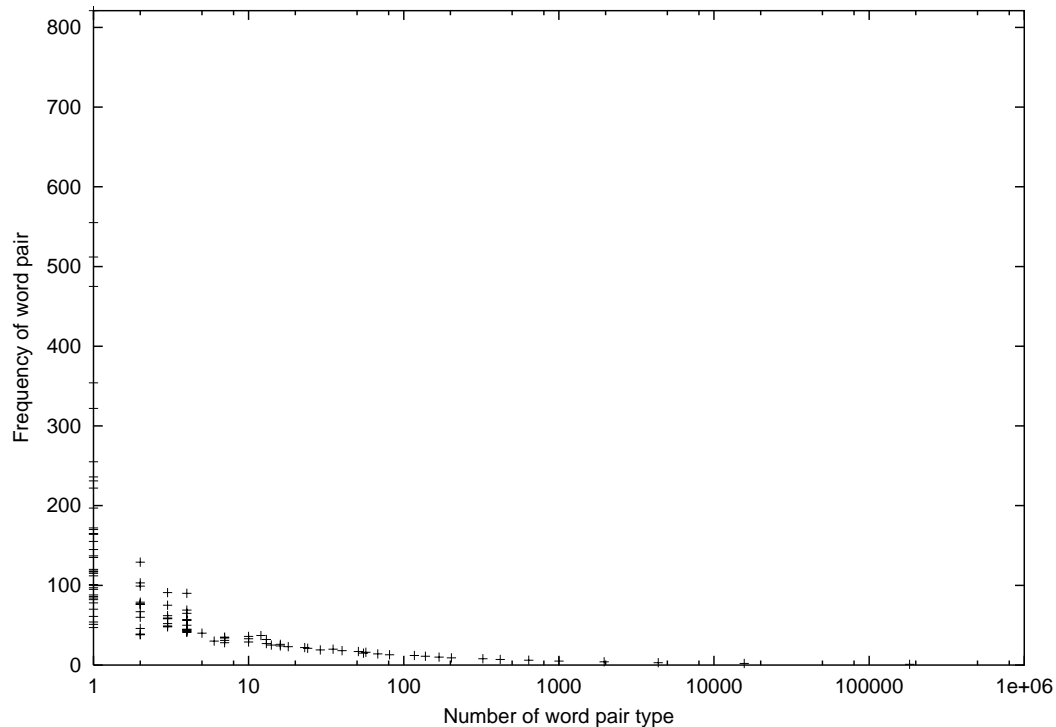


Figure 6.1: A distribution of word pairs. The graph shows the number of word types on the X-axis versus the frequency of the word pairs on the Y-axis. 182,479 word pairs among 208,233 pairs appear only once in the training set

from 27,694 training sentences. Figure 6.1 is the result. As expected, many lexical word pairs (87.63%) occur only once. Table 6.1 is the part of word pairs that appeared only once in the corpus. A straightforward example of the lexical dependency sparseness is that a word pair ‘밥을(*bap-eul*) 먹 (*meog*)’² appears only once. It is striking that this commonly using expression appears only once in the corpus. So it is doubtful whether this information can reflect selectional preference of words or not.

²means *have a meal*

word pairs (morpheme ternary)	meaning
폭동 (<i>pokdong</i>) 이(- <i>i</i>) 일어나 (<i>ireona</i>)	a riot aroses
포로 (<i>poro</i>) 를 (- <i>reul</i>) 학대하 (<i>hakdaeha</i>)	be cruel to a prisoner
포도주 (<i>podoju</i>) 를 (- <i>reul</i>) 담그 (<i>damgeu</i>)	brew wine
폐인 (<i>pyein</i>) 이 (- <i>i</i>) 되 (<i>doe</i>)	be crippled
평화 (<i>pyenghwa</i>) 를 (- <i>reul</i>) 유지하 (<i>yujiha</i>)	keep peace
평생 (<i>pyeongsaeng</i>) 을 (- <i>eul</i>) 바치 (<i>bachi</i>)	devote one's lifetime
편지 (<i>pyeonjireul</i>) 를 (- <i>reul</i>) 교환 (<i>gyohwanha</i>)	exchange letters
편견 (<i>pyongyeon</i>) 을 (- <i>eul</i>) 가지 (<i>haji</i>)	have a prejudice
판결 (<i>pangyeol</i>) 을 (- <i>eul</i>) 하 (<i>ha</i>)	give a decision
탄성 (<i>tanseong</i>) 을 (- <i>eul</i>) 지르 (<i>jireu</i>)	heave a sigh
코 (<i>ko</i>) 가 (- <i>ga</i>) 막히 (<i>makhi</i>)	nose is blocked
측면 (<i>cheungmyeon</i>) 에서 (- <i>eseo</i>) 다루(<i>daru</i>)	take a side view of
총탄 (<i>chongtan</i>) 을(- <i>eul</i>) 맞(<i>mat</i>)	by hit by a bullet
체제 (<i>cheje</i>) 의 (- <i>ui</i>) 붕괴(<i>bunggoe</i>)	the breaking of a system
청중 (<i>cheongjung</i>) 에게(- <i>ege</i>) 호소하(<i>hosoha</i>)	appeal to the listners
천하 (<i>chenha</i>) 를 (- <i>reul</i>) 통일하 (<i>tongilha</i>)	unify a country
책 (<i>chaeg</i>) 을 (- <i>eul</i>) 펴 (<i>pyeo</i>)	open a book
채널 (<i>channel</i>) 을 (- <i>eul</i>) 바꾸 (<i>bakku</i>)	change a channel
창피 (<i>changpi</i>) 를 (- <i>reul</i>) 당하 (<i>dangha</i>)	be put to shame
밥 (<i>bap</i>) 을 (- <i>eul</i>) 짓 (<i>jit</i>)	cook a meal
밥 (<i>bap</i>) 을 (- <i>eul</i>) 먹 (<i>meog</i>)	have a meal
반란 (<i>ballan</i>) 이(- <i>i</i>) 일어나(<i>ireona</i>)	a rebellion breaks out

Table 6.1: List of word ternaries that appeared only once in 27,694 sentences

6.1.3 Unlexicalized Parser

The lexicalized parsing model introduced in the previous chapter is modified to the unlexicalized parsing model. Formally,

$$\begin{aligned}
 P(T|S) &\approx \prod_{0 < i < |S|} P(dep_i|S) & (6.1) \\
 &\approx \prod_{0 < i < |S|} P(head(t_i) = t_{h(i)}, length(w_i) = \Psi(h(i) - i) | t_i t_{h(i)} \Phi_i \Phi_{h(i)}) \\
 &\approx \prod_{0 < i < |S|} P(head(t_i) = t_{h(i)} | t_i t_{h(i)} ct_{i-1} ft_{h(i)+1}) \\
 &\quad \cdot P(length(w_i) = \Psi(h(i) - i) | t_i \Phi_i)
 \end{aligned}$$

That is, an input sentence is assumed as a sequence of part-of-speech tags, not words :

$$S = t_1 t_2 t_3 \cdots t_{|S|} \quad (6.2)$$

The unlexicalized parser is tested on the identical training and testing data is used in the previous chapter. Table 6.2 shows the comparison of the parsing performance between the lexicalized parser of the previous chapter and the unlexicalized parser suggested in this chapter.

Lexicalized parser does extremely well for the training data because it memorizes the lexical dependency relation when learning and applies it untouched in testing. Unlexicalization makes the memorizing less effective. For testing data, the performance of unlexicalized parser is 1% of F -score worse than the lexicalized one. This figure is slightly larger (+0.5%) than that reported by Gildea (2001), which compares the unlexicalizing effect in English parsing.

When comparing other lexicalized parsers, such as **Model 4** from Collins (1996), in the previous chapter, the proposed unlexicalized parser is much better than them with unseen testing data (Figure 6.2). In addition, our unlexicalized parser required much smaller size of frequency data for estimating the probabilities. While the lexicalized parser of **Model4** requires 643M bytes for storing the data, our unlexicalized parser uses only 18M bytes of data. We didn't try to optimize the data structure. But taking that into account, the

	Measures	Lexicalized ($K = 4$)	Unlexicalized ($K = 4$)
Training Set	Arc F_1	0.9804	0.8606
	ExactMatch	0.8333	0.3548
Testing Set	Arc F_1	0.8674	0.8562
	ExactMatch	0.3404	0.3225

Table 6.2: Comparison of the parsing performance between the lexicalized and unlexicalized parser

	Measures	Model4 (Collins 1996)	Unlexicalized ($K = 4$)
Training Set	Arc F_1	0.9962	0.8606
	ExactMatch	0.9665	0.3548
Testing Set	Arc F_1	0.8370	0.8562
	ExactMatch	0.2569	0.3225

Table 6.3: Comparison of the parsing performance between the lexicalized parser (**Model4**) and our unlexicalized parser

huge difference of the required resource size gives some ideas why unlexicalized parser is preferable.

6.2 Parsing with Splitted Part-of-speech tags

6.2.1 Word Clustering

In dependency analysis, syntactic disambiguation depends much on the characteristic of words itself, rather than other factors, because the dependency grammar does not allow any intermediate structure between word and sentence. Therefore, lexical information may be more important in parsing with dependency grammar because the lexical form of a word

usually represents the characteristic of the word. However, if part-of-speech tags are fine-grained enough to reflect the words' syntactic-semantic property, the characteristic of words can be considered without using lexical form of words. Here, we deal with a way to split the existing part-of-speech tag set according to syntactic feature of the words that have the tag.

Part-of-speech tag splitting is similar to word clustering in respect that clustered words are assigned a tag, or a class, that represents the cluster. They both address issues of data sparseness and generalization in statistical language processing. One way to cluster words is to build clusters from preexisting classes constructed by humans. Resnik (1992) uses WordNet, and Yarowsky (1992) works with Roget's thesaurus. Another way is to derive classes directly from distributional data (Lee 1997b).

Usually, word clustering is a technique for partitioning sets of words into subsets of *semantically* similar words. For example, work of Pereira, Tishby, and Lee (1993) cluster words according to their distribution in particular syntactic contexts; They cluster noun according to their distribution as direct objects of verbs. Then the semantically similar words are collected in a cluster. edible nouns in a cluster, nouns related to organization in another cluster, etc.

The approach proposed here is somewhat different. We propose a word clustering technique based on *syntactic* property of the word; not *semantic* property. We assumed each word has its own preference on modification distance in the previous section. Based on the syntactic preference, we classify words into two groups, which are group prefers short modification and group prefers long modification.

Let's consider four nouns³ for example : 국가 (*gukga*; nation), 사회 (*sahoe*; society), 경우 (*gyeongu*; case), 결과 (*gyeolgwa*; result). We investigate the distribution of the length of dependency relation from those words when they are used as a right most word of a modifier. Results are shown on Figure 6.2 and 6.3.

As Figure 6.2 shows, the first two nouns absolutely prefer modifying next words or near words from them. On the other side, two words 경우 (*gyeongu*; case) and 결과 (*gyeolgwa*; result) modifies words apart from them frequently (Figure 6.3). Unlexicalizing the parsing

³Their actual part-of-speech tags are *ncn*.

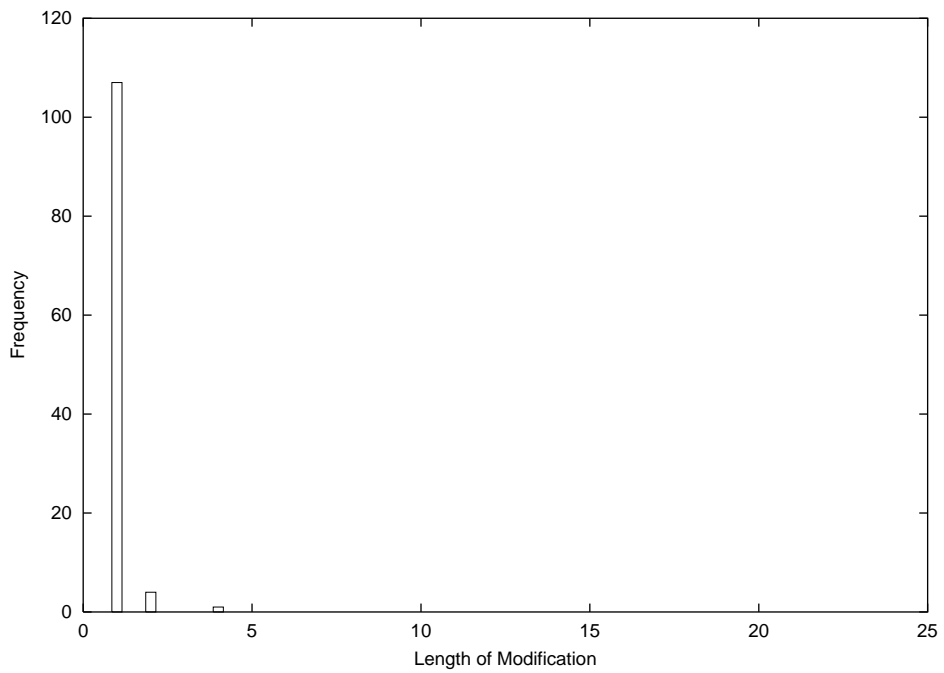
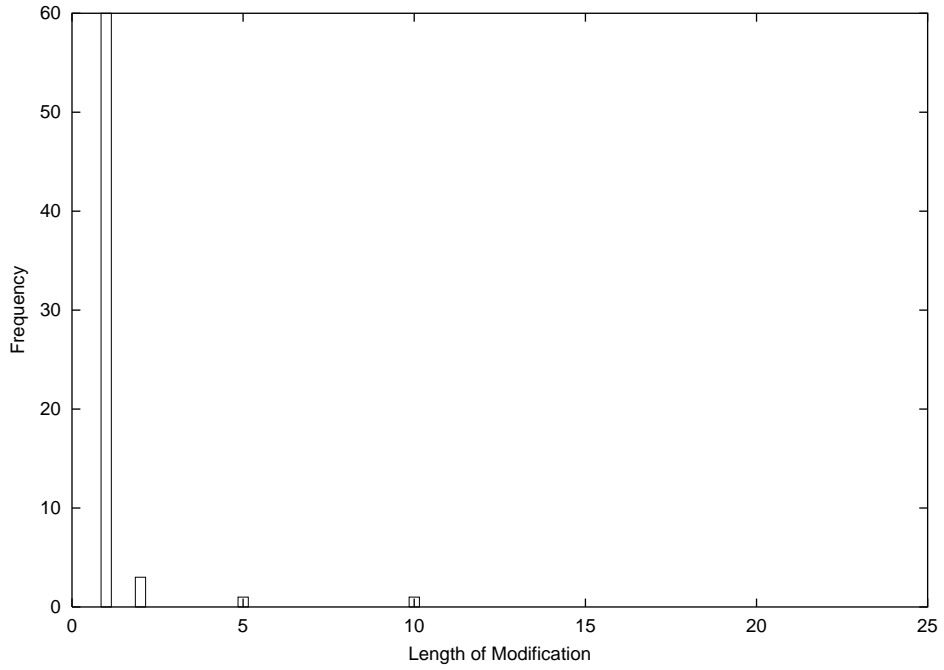


Figure 6.2: Distribution of the length of dependency relation from 국가 (*gukga*; nation) and 사회 (*sahoe*; society)

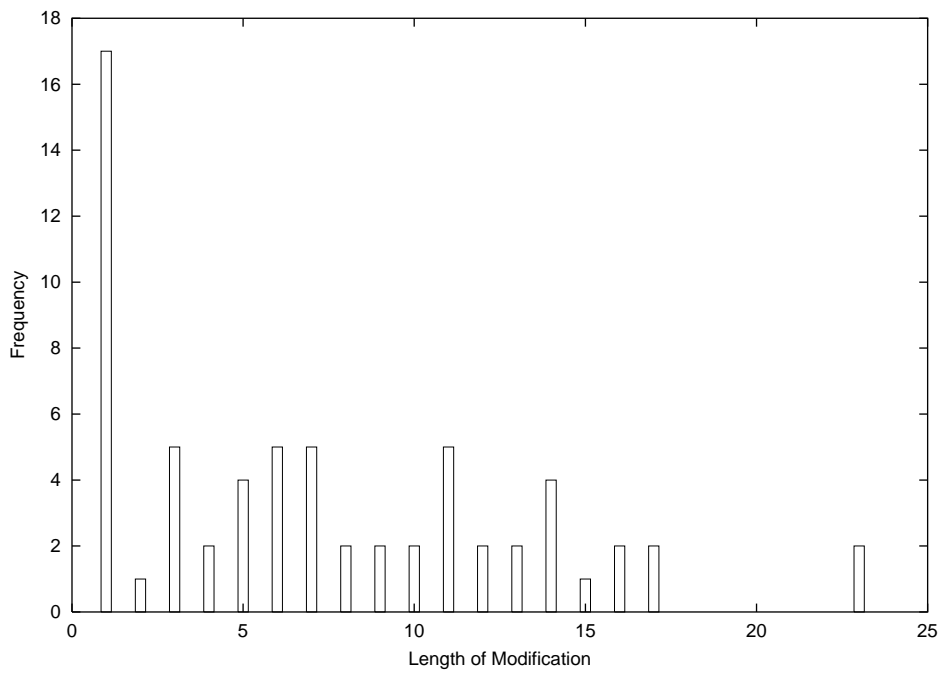
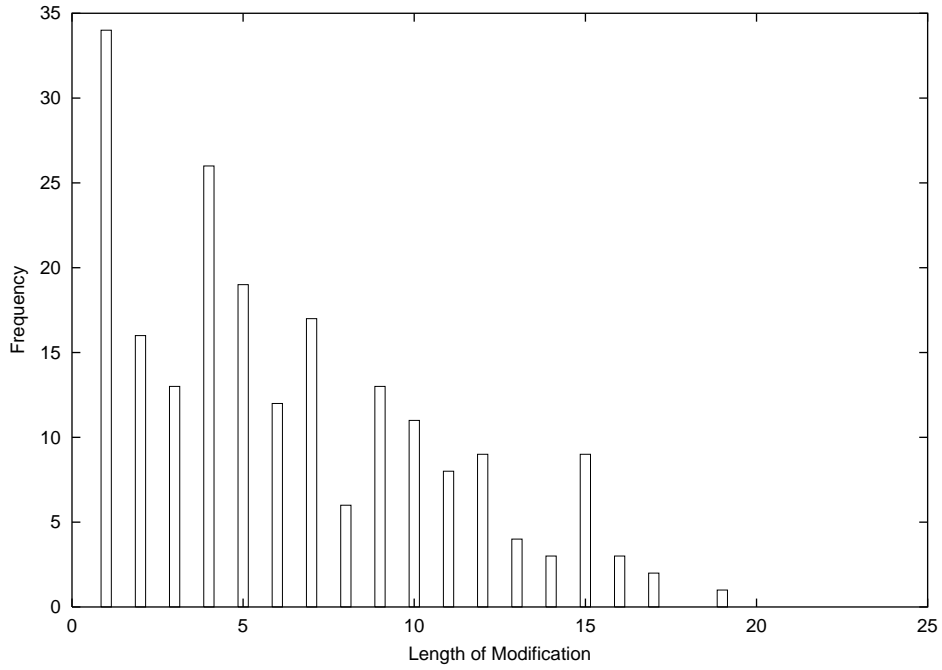


Figure 6.3: Distribution of the length of dependency relation from 경우 (*gyeonggu*; case) and 결과 (*gyeolgwa*; result)

model without considering this preference of word may result poorer performance because the modification distance probability in the parsing model would not be assigned properly.

A simple approach is used to classify a word according to the distribution of modification length it prefers.

$$word\ class = argmax_{long,short} P(length(w_i) = \Gamma(h(i) - i) | w_i) \quad (6.3)$$

$$\Gamma(d) = \begin{cases} short & \text{if } i \leq K \\ long & \text{otherwise} \end{cases}$$

where K is a yardstick introduced in the previous chapter. It is used to decide whether a modification length is considered to be long or short. According to the classifier, 국가 (*gukga*; nation) and 사회 (*sahoe*; society) become to have part-of-speech tag **ncn1** (**ncn** preferring short modification distance) while 경우 (*gyeongu*; case) and 결과 (*gyeolgwa*; result) become to have **ncn0** tag (**ncn** preferring long modification distance).

If an unknown word appears in an input sentence, splitted part-of-speech tag cannot be assigned to the word, because the word is not in the splitted tag dictionary. In this case, we assign a most frequent tag for the word. For instance, a new word 컴퓨터 (computer), whose part-of-speech tag is **ncn**, will get **ncn0** splitted tag, because **ncn0** is more frequent than **ncn1** for the part-of-speech **ncn**.

Six part-of-speeches are used in clustering experiment: **ecs**, **mad**, **mag**, **maj**, **nbn**, and **ncn** and Table 6.4 is the result.

6.2.2 Parsing With Splitted Tag

We splitted tag with word clusters To make up for the performance decrease comes from unlexicalization Using splitted tags instead of part-of-speech tags, the performance of parser changes as Table 6.5. The parser using splitted tag information does better than purely unlexicalized parser.

To analyze the contribution of tag splitting in performance changes, we compare accuracies of parsers that use lexical/splitted tag/unlexical information in measuring word dependency probability and modification distance probability. The performance of those

Tag	Splitted Tag	Word
ecs	ecs0	-ㄴ 다면 -ㄴ지라도 -나 는다면 -니 -다면 -더니 -더라도 -더라면 -듯이 -라도 -라면 -려면 -면 -므로 -으니 -지만 ... (Total 28)
	ecs1	-ㄴ 다 -ㄴ 지 -ㄴ 수록 -고서 -고자 -는지 -니까 -다 -다가 -다시피 -도록 -려고 -면서 -어야 -으려고 -으면 -자마자 ... (Total 35)
mad	mad0	이때 (Total 1)
	mad1	그렇게 그리 이렇게 (Total 3)
mag	mag0	가령 간혹 결국 그때 다만 다시금 달리 대개 대부분 대체로 더구나 더욱이 또한 마침내 만약 만일 말하자면 한편 ... (Total 38)
	mag1	가까이 가끔 가득 가만히 가장 각각 각기 간단히 갑자기 거꾸로 거듭 거의 겨우 결코 계속 고루 곧 곧바로 과연 굳이 그나마 그냥 그다지 그대로 그럼에도 그만큼 그야말로 그저 그토록 극히 금방 깜짝 깨끗이 꼭 꽤 끝내 내내 너무 단지 대단히 더 또다시 비록 빨리 ... (Total 163)
maj	maj0	그래도 그래서 그러나 그러니 그러니까 그러면 그러므로 그러자 그런데 그럼 그렇다면 그렇지만 그리하여 드디어 ... (Total 27)
	maj1	곧 내지 또는 및 혹은 (Total 5)
nbn	nbn	동안 때 무렵 한편 (Total 4)
	nbn	-내 -대로 -데 -듯 -등 -때문 -만큼 -바 -분 -뿐 -자 -점 -줄 -증 -지 -채 -편 -한 -후 (Total 20)
ncn	ncn0	결과 경우 다음날 동안 때 반면 이때 이래 이제 이후 직후 한때 후일 (Total 14)
	ncn1	EC TV 가격 가구주 가슴 가운데 가족 가치 각종 강도 갖가지 개념 개인 거리 건물 게 겨울 견해 경기 경상도 경제 계급 고대 곳 공 공간 공산주의 공업 공화국 과거 과정 과제 과학 관념론 국가 국내 국민 군대 군사 권위 ... (Total 496)

Table 6.4: Result of part-of-speech tag splitting

	Measures	Lexicalized	Unlexicalized	Splitted Tag
Training Set	Arc F_1	0.9804	0.8606	0.8774
	ExactMatch	0.8333	0.3548	0.3826
Testing Set	Arc F_1	0.8674	0.8562	0.8633
	ExactMatch	0.3404	0.3225	0.3357

Table 6.5: Comparison of the parsing performance between the lexicalized and unlexicalized parser

	WordDep Unlex	WordDep Spl. Tag	WordDep Lex
ModDist Unlex	0.8562 (0.3225)	0.8598 (0.3307)	0.8658 (0.3349)
ModDist Spl.Tag	0.8602 (0.3278)	0.8633 (0.3357)	0.8685 (0.3396)
ModDist Lex	0.8582 (0.3225)	–	0.8675 (0.3405)

Table 6.6: Parsing with combination of unlexicalized, splitted tagged, lexicalized information on the testing data

parsers in testing set are shown in Figure 6.6. It seems the splitted tag information does not help much in disambiguation. An interesting point is that using splitted tag is better than using lexical word in measuring modification distance probability. And this results the parser using splitted information in estimating modification distance probability and lexical information in lexical dependency probability performs a little bit better than lexicalized parser (+0.1% absolute).

Chapter 7

Conclusion

This dissertation has proposed a new probabilistic model for parsing the Korean language. In the proposed parsing model, preference for a modification distance in a certain local context is considered in addition to the preference for lexical bigram dependency. All of these preferences are expressed by probabilities conditioned on local context. The lexical dependency probability reflects lexical bigram dependency, selectional preference and the preference on each dependency rules. The modification distance probability reflects the preferred length of a dependency relation from a certain dependent word based on the context of the dependent word. The parsing model is based on the dependency theory, which is widely known as an adequate formalism to reflect the syntactic characteristic of the Korean language, or other variable word-order languages.

We designed the model based on the characteristics of the Korean language. Evaluation on the KAIST Treebank text showed that the proposed model recovered dependency relations with 86.75% F_1 -score.

We can conclude that

- The surface contextual pattern helps syntactic disambiguation even for the variable-word-order language
- The distance measure is useful for syntactic disambiguation for the variable-word order language.

- Splitting lexical dependency preference and modification distance preference is effective
- Modification distance depends on the modifier, rather than the modifyee.

7.1 Contributions

The following summarized the contributions of our work.

- A simple parsing model based on the lexical dependency probability or likelihood of dependency rule usage could not express a tendency on short (or long for some languages) distance dependency relations. However, the tendency is empirically proved to be useful in syntactic disambiguation. We proposed a model that formally considers the preference without ignoring grammatically correct dependency relations.
- We also showed that the features depends on word-order, such as surface contextual information, help disambiguating syntactic structure of free-word order languages.
- Furthermore, we proposed the way to design a model that is robust for data sparseness problem, considering the lexical preference and the preference on the length of dependency relation, at the same time.
- Though we designed the model that reflects the characteristic of Korean, it use little language dependent feature. Therefore, the proposed parsing model can be used for various languages which allow free word order.
- We used a public-available corpus for training and testing our parsing model. We also specified what parts of the corpus has been used as training and testing set. We believe collaborative competitions in the Korean language parsing can be induced by this kind of work.

7.2 Future works

For future work, we are planning to pursue the followings. First, unlexicalization parsing. Due to the sparse data of lexical dependencies, the parser with the proposed parsing model

requires huge amount of data storage space. This causes slow loading of the lexical data and complex smoothing techniques. Slimming the resources used in the parsing may help developing practical parser. Unlexicalized parser suggested in Discussion will be a starting point.

Secondly, we are likely to acquire lexical dependency information from raw corpora to increase parsing performance. The proposed model just uses the lexical bigram dependency from the Treebank, which is very small size. By using the automatically extracted lexical dependency information, the parser may perform better.

And we hope to carry out experiments with other languages such as Japanese. We believe the parsing model can be easily adapted to other languages, which have similar syntactic characteristics as Korean's.

Appendix A

Functional Morpheme

Abbreviation List

A set of abbreviations for the Korean functional morphemes is defined for explanatory purpose.

	Abbrev.	Full name	Corresponding Morpheme
Case	NOM	Nominative case marker	<i>-ga, -i</i>
Marker	ACC	Accusative case marker	<i>-eul, -reul</i>
	LOC	Locative case marker	<i>-e, -egseo</i>
	TMP	Temporal case marker	<i>-e</i>
	DAT	Dative case marker	<i>-ege</i>
	CONJ	Conjunctive case marker	<i>-wa, -gwa</i>
	TOP	Topical case marker	<i>-eun, -neun</i>
	GEN	Genitive case marker	<i>-ui</i>
	CMPL	Complemental case marker	<i>-ga, -i</i>
	COMP	Comparative case marker	<i>-boda</i>
	AUX	auxiliary case marker	<i>-do</i>

Verb	PAST	Past tense marker	<i>-at, -eot</i>
Ending	OBL	Obligatory marker	<i>-eoya</i>
	END	Final ending marker	<i>-da</i>
	SUBCONJ	Subordinate conjunctive ending marker	<i>-myun, -go, \dots</i>

Appendix B

Part-of-speech Tag Set

The part-of-speech tags used in this dissertation is from the KAIST Treebank (Lee, Kim, Chang, Choi, and Kim 1996) of the KAIST Language Resources (Choi 2001)¹. See (Choi, Nam, Kim, Han, Park, Kim, Lee, Kim, Kim, and Choi 1996) for further details on this tag set.

Category	Sub-Category	Tag	Description
symbol		sf	full stop
		sp	pause
		sl	left quotation and parenthesis mark
		sr	right quotation and parenthesis mark
		sd	dash
		se	ellipsis
		su	unit
		sy	other symbols
foreign		f	foreign word

¹We follow (Lee 1997a) for the English translation of the tag description.

Category	Sub-Category	Tag	Description
noun	common	nepa	active-predicative common noun
		ncps	statove-predicative common noun
		ncn	non-predicative common noun
	proper	nq	proper noun
	bound	nbu	unit bound noun
		nbn	non-unit bound noun
	pronoun	npp	personal pronoun
		npd	demonstrative pronoun
	numeral	nnc	cardinal numerals
		nno	ordinal numerals
predicate	verb	pvd	demonstrative verb
		pvg	general verb
	adjective	pad	demonstrative adjective
		paa	attributive adjective
	auxiliary	px	auxiliary verb
	modification	adnoun	mmd
mma			attributive adnoun
adverb		mad	demonstrative adverb
		maj	conjunctive adverb
		mag	general adverb
independence	interjection	ii	interjection

Category	Sub-Category	Tag	Description
particle	case	jcs	subjective case particle
		jco	objective case particle
		jcc	complemental case particle
		jcm	adnominal case particle
		jcv	vocative case particle
		jca	adverbial case particle
		jcj	conjunctive case particle
		jct	comitative case particle
		jcq	quotative case particle
	auxiliary	jxc	common auxiliary particle
jxf		final auxiliary particle	
predicative	jcp	predicative particle	
ending	prefinal	ep	prefinal ending
	conjunctive	ecc	coordinate conjunctive ending
		ecs	subordinate conjunctive ending
		ecx	auxiliary conjunctive ending
	transform	etn	nominalizing ending
		etm	adnominalizing ending
	final	ef	final ending
affix	prefix	xp	prefix
	suffix	xsn	noun-derivational suffix
		xsv	verb-derivational suffix
		xsm	adjective-derivational suffix
		xsa	adverb-derivational suffix

Appendix C

Simple AUXP Chunker

In Korean, a prefinal verbal ending or an auxiliary verb represent tense, honorific, aspect, and modal information. These prefinal verbal endings and the auxiliary verbs can construct the auxiliary phrases (AUXP), which can be attached on a verb or an adjective (Lee 1997a).

The parser proposed in this dissertation assumes the AUXP chunked input. Length of a dependency relation can be better indicator if AUXPs are reduced to a single morpheme. The AUXP prechunking is also helpful in reducing parsing complexity, because the process usually reduce the length of sentences and parsing complexity increases in proportion to the sentence length.

A primary set of AUXP chunking rules, which is based on part-of-speech tags and morphemes, is encoded with reference to the KAIST Treebank bracketing guideline (Lee, Chang, and Kim 1997). We applied the set of rules to the training corpus for the parser, and add new rules to the rule set to chunk AUXP that cannot be found by the primary rules. Table C.1 is a part of AUXP chunking rule set.

Experiments on the testing data reports recall/precision rate of 0.9999% and 0.9756% (Chung 2002).

AUXP -> ep
 AUXP -> ecx px
 AUXP -> ecx+jcs px
 AUXP -> ecx+jco px
 AUXP -> ecx+jxc px
 AUXP -> ecx+jxc+jxt px
 AUXP -> ecx+jxt px
 AUXP -> ecx+jxt+px
 AUXP -> ecx+px
 AUXP -> ecx nbn+jp
 AUXP -> etm nbn_것|셈|분|터|따름|지경|법|모양|거+jp
 AUXP -> etm nbn_것|셈|분|터|따름|지경|법|모양|거+xsn+jp
 AUXP -> etm nbn_수|박|데|리 paa_있|없
 AUXP -> etm nbn_수|박|데|리+jcs paa_있|없
 AUXP -> etm nbn_수|박|데|리+jxc paa_있|없
 AUXP -> etm nbn_수|박|데|리+jxt paa_있|없
 AUXP -> etm nbn_수|박|데|리+jxc+jcc paa_있|없
 AUXP -> etm nbn_수|박|데|리+jxc+jxt paa_있|없
 AUXP -> etm nbn_것 paa_갈
 AUXP -> etm nbn_것+jxc paa_갈
 AUXP -> etm nbn_거+jcc paa_갈
 AUXP -> etm nbn_거+jcs paa_갈

AUXP -> etm nbn_뿐 paa_아니
 AUXP -> etm nbn_뿐+jxc paa_아니
 AUXP -> etm nbn_뿐+jxt paa_아니
 AUXP -> etm nbn_뿐+jxc+jcc paa_아니
 AUXP -> etn_기 nbn_때문+jp
 AUXP -> etn_기|계 nbn_마련+jp
 AUXP -> etm_은|는|니|리|을|를 px
 AUXP -> etn_기+jxc px_아
 AUXP -> etn_기+jxc+jxc px_아
 AUXP -> etn_기+jxt px_아
 AUXP -> etn_기+jca px_아
 AUXP -> etn_기+jxc pvg_아

Figure C.1: AUXP chunking rule set

Bibliography

- Allen, J. (1995). *Natural Language Understanding*. Benjamin Cummings.
- Bien, J. S. and S. Szpakowics (1982). Toward a parsing method for free word order languages. In *Proceedings of 9th International Conference on Computational Linguistics (COLING)*.
- Black, E., A. Abney, D. Flickenger, C. Gdaniec, R. Grishman, P. Harrison, D. Hindle, R. Ingria, F. Jelinek, J. Klavans, M. Liberman, M. Marcus, S. Roukos, B. Santorini, and T. Strzalkowsk (1991). A procedure for quantitatively comparing the syntactic coverage of english grammars. In *Proceedings of Fourth DARPA Speech and natural Language Workshop*.
- Charniak, E. (1993). *Statistical Language Learning*. The MIT Press.
- Charniak, E. (1995). Parsing with context-free grammar and word statistics. Technical Report CS-95-28, Department of Computer Science, Brown University.
- Charniak, E. (1996). Tree-bank grammars. Technical Report CS-96-02, Department of Computer Science, Brown University.
- Charniak, E. (1999). A maximum-entropy-inspired parser. Technical Report CS-99-12, Department of Computer Science, Brown University.
- Charniak, E. (2001). Immediate-head parsing for language models. In *Meeting of the Association for Computational Linguistics*, pp. 116–123.
- Chelba, C. and M. Mahajan (2001). Information extraction using the structured language model. In *Proceedings of EMNLP/NAACL 2001*.

- Chen, J., A. Diekema, M. D. Taffet, N. J. McCracken, N. E. Ozgencil, O. Yilmazel, and E. D. Liddy (2001). Question answering: CNLP at the TREC-10 question answering track. In *The 10th Text REtrieval Conference*.
- Choi, K.-S. (2001). KAIST Language Resources v.2001. Result of Core Software Project from Ministry of Science and Technology, Korea(<http://kibs.kaist.ac.kr>).
- Choi, K.-S., Y. Nam, J. Kim, Y. Han, S. Park, J. Kim, C. Lee, D. Kim, J.-H. Kim, and B. Choi (1996). A study of the morphological and syntactic tag standard for korean language information bases. *Korean Journal of Cognitive Science* 7(4).
- Chung, H. (2002). AUXP chunking. Internal Report 020612, Natural Language Processing Lab. Korea University.
- Collins, M. (1999). *Head-Driven Statistical Models for Natural Language Parsing*. Ph. D. thesis, University of Pennsylvania.
- Collins, M. J. (1996). A new statistical parser based on bigram lexical dependencies. In *Proceedings of the 34th Annual Meeting of the ACL*.
- Collins, M. J. (1997). Three generative, lexicalised models for statistical parsing. In *Proceedings of the 35th Annual Meeting of the ACL*.
- Covington, M. A. (1990). A dependency parser for variable-word-order languages. Research Report AI-1990-01, Artificial Intelligence Programs, The University of Georgia.
- Debusmann, R. (2000). An introduction to dependency grammar.
- Eisner, J. M. (1998). Three new probabilistic models for dependency parsing: an exploration. In *Proceedings of the 16th International Conference on Computational Linguistics (COLING-96)*.
- Fox, H. J. (2002). Phrasal cohesion and statistical machine translation. In *Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing (EMNLP 2002)*, pp. 304–311.
- Geum, J. C. and G. Kim (1988). Implementation of HPSG parsing mechanism for Korean syntactic structure analysis. In *Proceedings of the Spring Conference of Korea Information Science Society*, pp. 139–142.

- Gildea, D. (2001). Corpus variation and parser performance. In *Proceedings of Conference on Empirical Methods in Natural Language Processing*.
- Jung, H.-S., J.-H. Kim, J.-S. Lee, S.-Y. Chun, and M.-J. Park (1989). Design of Korean-English machine translation system (KoEng). In *Proceedings of the 1st Workshop of Machine Translation*, pp. 87–96.
- Jung, S.-W., E.-K. Park, D.-Y. Ra, and J.-T. Yoon (2001). A study on Korean dependency parser using case relation and mutual information. In *Proceedings of the Hangul and Korean Information Processing Conference*.
- Kanayama, H., K. Torisawa, Y. Mitsuichi, and J. Tsujii (1999). Statistical dependency analysis with an HPSG-based Japanese grammar. In *Proceedings of 5th Natural Language Processing Pacific Rim Symposium*, pp. 138–143.
- Kanayama, H., K. Torisawa, Y. Mitsuichi, and J. Tsujii (2000). A hybrid Japanese parser with an hand-crafted grammar and statistics. In *Proceedings of the 18th International Conference on Computational Linguistics (COLING-2000)*, pp. 411–417.
- Kim, C.-H. (1993). A right-to-left chart parser for Korean. Master’s thesis, Dept. of Computer Science. KAIST.
- Kim, C.-H., J.-H. Kim, and J. Seo (1993). A right-to-left parsing using headable path. In *Proceedings of the 5th Hangul and Korean Information Processing Conference*.
- Kim, H. (1994). Korean syntactic analysis with probabilistic dependency grammar. Master’s thesis, Dept. of Computer Science. KAIST.
- Kim, H. and J. Seo (1997). A statistical Korean parser based on lexical dependencies. In *Spring Proceedings of Conference on Korea AI Society*.
- Kim, J., H. Kim, and J. Seo (1997). Dependency structure analysis system for Korean sentences using statistical methods. In *Proceedings of the Spring Conference of Korea Cognitive Science Society*, pp. 200–209.
- Kim, K.-B., E.-K. Park, D.-Y. Ra, and J.-T. Yoon (2002). A method of Korean parsing based on sentence segmentation. In *Proceedings of the 14th Hangul and Korean Information Processing Conference*.

- Kim, M.-Y., S.-J. Kang, and J.-H. Lee (2001). Resolving ambiguity in inter-chunk dependency parsing. In *Proceedings of NLPRS 2001*, pp. 263–270.
- Klein, D. and C. D. Manning (2003). Accurate unlexicalized parsing. In *Proceedings of the 41st Meeting of the Association for Computational Linguistics*, pp. 310–315.
- Kruijff, G.-J. M. (2002). Formal & computational aspects of dependency grammar : Heads, dependents, and dependency structures. <http://www.coli.uni-sb.de/~gj/Lectures/DG.ESSLLI/index.phtml>.
- Kwak, Y.-J., S.-Y. Park, Y.-S. Hwang, H. Chung, S.-Z. Lee, and H.-C. Rim (2003). Generalized LR parser with Conditional Action Model(CAM) using surface phrasal types. *Journal of Korea Information Science Society* 30(2(B)), 81–92.
- Kwon, H.-C. and J.-Y. Choi (1992). A Korean language parser with a unification-based dependency grammar. *Journal of Korea Information Science Society* 19(5), 467–476.
- Lee, J. and Y. Bae (1987). *Linguistics Dictionary*. Parkyoungsa.
- Lee, K. J. (1997a). *Probabilistic Parsing of Korean based on Language-Specific Properties*. Ph. D. thesis, Dept. of Computer Science. KAIST.
- Lee, K. J., B.-G. Chang, and G. Kim (1997). Bracketing guidelines for korean syntactic tree tagged corpus : Version 1. Technical Report CS/TR-97-112, Department of Computer Science, KAIST.
- Lee, K. J., J.-H. Kim, B.-G. Chang, K.-S. Choi, and G. C. Kim (1996). Syntactic tag set for korean syntactic tree tagged corpus. Technical Report Report CS/TR-96-102, Department of Computer Science, KAIST.
- Lee, K. J., J.-H. Kim, and G. C. Kim (1996). Probabilistic parsing with Korean phrase structure grammar. In *Proceedings of the Fall Conference of Korea Information Science Society*.
- Lee, K. J., J.-H. Kim, and G. C. Kim (1998). Syntactic analysis of Korean sentences based on restricted phrase structure grammar. *Journal of Korea Information Science Society* 25(4).

- Lee, L. J. (1997b). *Similarity-Based Approaches to Natural Language Processing*. Ph. D. thesis, Harvard University.
- Lee, S. (2002). *A Statistical Model for Identifying Grammatical Relations in Korean Sentences*. Ph. D. thesis, Dept. of Computer Science. Sogang University.
- Magerman, D. (1995). Statistical decision-tree models for parsing. In *Proceedings of the 33rd Annual Meeting of the Association for Computational Linguistics*.
- Miller, S., H. Fox, L. Ramshaw, and R. Weischedel (2000). A novel use of statistical parsing to extract information from text. In *Proceedings of the NA-ACL*.
- of Culture & Tourism, M. (2000). The revised romanization of Korean : Notification no. 2000-8 of Ministry of Culture & Tourism, Republic of Korea.
- Park, S.-Y., Y.-S. Hwang, and H.-C. Rim (1999). A morpheme-unit Korean Feature-based Grammar (KFG) with the X-bar theoretic notion of headedness. *Journal of Korea Information Science Society* 26(10(B)), 1247–1259.
- Pereira, F., N. Tishby, and L. Lee (1993). Distributional clustering of English words. In *The 31st Annual Meeting of the Association for Computational Linguistics*, pp. 183–190.
- Ra, D.-Y. (1994). A study on Korean parsing. *Communications of the Korea Information Science Society* 12(8), 33–46.
- Resnik, P. (1992). Wordnet and distributional analysis : A class-based approach to lexical discovery. In *Proceedings of AAAI Workshop on Statistically-based Natural Language Processing Techniques*.
- Ryu, P., J.-H. Lee, and G. Lee (1996). Using local dependency for dependency parser of Korean. In *Proceedings of the Hangul and Korean Information Processing Conference*.
- Sekine, S., K. Uchimoto, and H. Isahara (2000). Backward beam search algorithm for dependency analysis of Japanese. In *Proceedings of the 18th International Conference on Computational Linguistics (COLING-2000)*, pp. 745–760.
- Seo, K.-J. (1993). A Korean language parser using syntactic dependency relations between word-phrases. Master’s thesis, Dept. of Computer Science. KAIST.

- Seo, K.-J. (1998). *Probabilistic Dependency Parsing for Korean based on Ascending Dependency*. Ph. D. thesis, Dept. of Computer Science. KAIST.
- Seo, K.-J., K.-C. Nam, and K.-S. Choi (1999). A probabilistic model of the dependency parse for the variable-word-order languages by using ascending dependency. *Computer Processing of Oriental Languages* 12(3), 309–322.
- Seo, Y.-H. and Y. T. Kim (1990). Parsing of partially-free word order Korean sentences with head-final concept. In *Proceedings of the International Conference on Computer Processing of Chinese and Oriental Language*, pp. 47–49.
- Woo, S.-G. (1992). Korean syntactic analysis with syntactic relation. Master’s thesis, Dept. of Computer Science. KAIST.
- Xu, J., A. Licuanan, J. May, S. Miller, and R. Weischedel (2002). TREC 2002 QA at BBN: Answer selection and confidence estimation. In *The 11th Text REtrieval Conference*.
- Yamada, K. and K. Knight (2001). A syntax-based statistical translation model. In *Proceedings of the Association for Computational Linguistics (ACL)*.
- Yang, J. (1990). A study on the Korean analyzer based on HPSG. Master’s thesis, Dept. of Computer Engineering. Seoul National University.
- Yarowsky, D. (1992). Word-sense disambiguation using statistical models of Roget’s categories trained on large corpora. In *Proceedings of the 15th International Conference on Computational Linguistics (COLING-92)*.
- Yoon, D. H. (1990). A parsing mechanism using cell elimination in dependency structure grammar. In *Proceedings of the Seoul International Conference on Natural Language Processing*, pp. 156–161.
- Yoon, D. H. (1993). *Korean Parsing using Idiomatic Information*. Ph. D. thesis, Dept. of Computer Engineering. Seoul National University.
- Yoon, D. H. and Y. T. Kim (1989). Analysis techniques for Korean sentence based on Lexical Functional Grammar. In *Proceedings of the International Parsing Workshop ’89*, pp. 369–378.

Yoon, D. H. and Y. T. Kim (1992). The Korean language analysis algorithm based on the dependency grammar using the multi-phase filtering and searching method. *Journal of Korea Information Science Society* 19(6), 614–624.

Yoon, J. (1997). *Syntactic Analysis for Korean Sentences Using Lexical Association Based on Co-occurrence Relation*. Ph. D. thesis, Dept. of Computer Science. Yonsei University.

요약문

자연어 파싱은 자연어처리기술을 요구하는 많은 작업 분야에서 해결해야 할 중요한 문제이다. 언어를 처리하는 많은 작업에서는 술어-논항 관계나 수식어-피수식어 관계에 대한 정보를 이용하는데, 파싱은 문장의 단어나 구 사이의 관계를 파악해 줌으로써 이 정보의 추출을 가능하게 한다. 그러나 자연어 문장이 내포하고 있는 중의성 때문에 문장을 정확하게 파싱하는 것은 어려운 일이다. 최근 십년 동안 통계적인 방법이 자연어 파싱, 혹은 통사적 중의성 해소에 널리 사용돼왔다. 통계적 자연어 파싱에서 가장 중요한 두 가지 일은 통사적 중의성 해소에 유용한 자질을 선택하는 것과 이들을 이용한 통계적인 모형을 설계하는 것이다.

본 논문에서는 수식 거리와 지역 문맥과 같은 표층 문맥 정보가 한국어의 통사적 중의성 해소에 미치는 영향을 논하며, 어휘 바이그램 의존 선호도 및 특정 지역 문맥에서의 수식 거리 선호도를 고려한 파싱 모형을 제안한다. 이들 선호도는 지역 문맥에 대한 조건부 확률로 표현된다. 이 파싱 모형은 의존 이론에 기반한 모형인데, 이는 한국어와 같은 자유어순언어의 통사적 특징을 반영하는데 적합한 이론이다. 제안하는 통계적 의존 파싱 모형은 어휘 의존 확률과 수식 거리 확률의 두 확률로 이루어지는데, 어휘 의존 확률은 의존 규칙 선호도 및 선택 선호도를 반영하고, 수식 거리 확률은 수식어의 주위 문맥이 주어졌을 때, 수식어로부터 시작되는 의존 관계의 길이에 대한 선호도를 반영한다.

어떤 언어에 대한 파싱 모형의 패러미터 형태를 정의할 때에는 고려하는 언어의 특징을 잘 반영하여야만 한다. 수식 거리에 대한 확률은 한국어와 같은 자유어순 언어의 특징을 고려하여 정의되었는데, 이는 의존하는 두 단어 사이의 거리를 반영하는 새로운 방법이다. 제안하는 모형은 KAIST 트리뱅크를 이용한 평가에서 의존 관계 단위 F_1 기준으로 86.75%의 성능을 보였다. 자유어순 언어에서도 수식 거리와 지역 문맥을 고려하는 것이 수식어의 올

바른 지배소를 선택하는 데 도움이 됐고, 제안한 방법으로 주식 거리를 고려함으로써 주식 거리를 고려하는 다른 통계적 모형들보다 더 나은 성능을 얻을 수 있었다.

감사의 글

먼저 이 모든 것을 가능하게 해주신 하나님께 감사와 영광을 돌립니다. 그리고 학부 시절부터 지금까지, 무려 11년 동안 저의 지도교수님이셨던 임해창 교수님께 깊은 감사를 전합니다. 부족한 저에게 자상한 지도와 인자한 사랑을 아끼지 않으신 것에 대해 진심으로 감사드립니다. 바쁘신 가운데에도 논문의 심사를 기꺼이 맡아주신 이성환 교수님, 육동석 교수님, 남기춘 교수님, 김현철 교수님께도 감사를 드립니다.

항상 저에게 모범을 보여주셨던, 우리 자연어처리 연구실을 떠난 모든 선배님들께 감사드립니다. 동기 원호와 많은 연구실 후배들에게도 감사드립니다. 이분들이 없었다면 7년의 연구실 생활을 견뎌내지 못했겠죠. 제가 연락을 잘 하지 않아도 꾸준히 저에게 관심을 보여주고 연락을 취해준 대학, 고등학교, 통신동호회 선배, 후배, 친구들에게도 감사드립니다.

마지막으로 저와 항상 함께 했던 부모님과 동생 후경이, 그리고 제가 대학원에 입학하면서 만나기 시작하여 지금은 제 아내가 된 성림이에게 깊은 감사를 드립니다.