PAPER Special Issue on Development of Advanced Computer Systems

Normalizing Syntactic Structures using Part-of-Speech Tags and Binary Rules

Seongyong KIM†, ***Student Member***, **Kong-Joo LEE**††, ***and*** **Key-Sun CHOI**†, ***Nonmembers***

SUMMARY We propose a normalization scheme of syntactic structures using a binary phrase structure grammar with composite labels. The normalization adopts binary rules so that the dependency between two sub-trees can be represented in the label of the tree. The label of a tree is composed of two attributes, each of which is extracted from each sub-tree, so that it can represent the compositional information of the tree. The composite label is generated from part-of-speech tags using an automatic labelling algorithm. Since the proposed normalization scheme is binary and uses only part-of-speech information, it can readily be used to compare the results of different syntactic analyses independently of their syntactic description and can be applied to other languages as well. It can also be used for syntactic analysis, which performs higher than the previous syntactic description for Korean corpus.

We implement a tool that transforms a syntactic description into normalized one based on this proposed scheme. It can help construct a unified syntactic corpus and extract syntactic information from various types of syntactic corpus in a uniform way.

key words: *normalization of syntactic structures, Korean, binary phrase structure grammar, composite label, automatic labelling algorithm*

1. Introduction

Syntactic analysis of a language is the process of mapping surface structures of a language into some tree-like structures. The basic task for syntactic analysis is to define part of speech (POS) tags of a language, syntactic categories to be used, and rules in a certain format with a general rule description to cover the characteristics of the language. Since each corpus and its corresponding syntactic analyzer use their own definitions of the above, we encounter the normalization problem in comparing the outputs of different syntactic analyzers.

Fig. 1 depicts two different syntactic trees for the same Korean input sentence. Fig. 1(a) is built using phrase structure grammar (PSG) according to Penn Korean Treebank [1] whereas Fig. 1(b) is done using restricted phrase structure grammar (RPSG) [2] in accordance with KAIST corpus [3].

The both use different sets of POS tags as well as different syntactic categories. In addition, the unit of bracketing in PSG is a word phrase, while it is a morpheme in RPSG. As a whole, because the both use

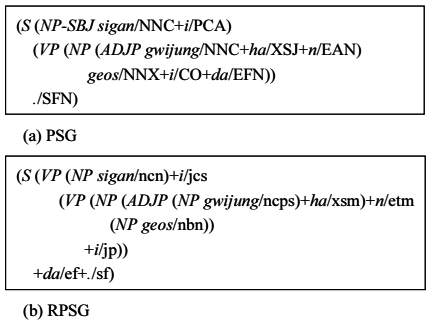


Fig. 1 Two syntactic structures for the same Korean sentence

different grammar specifications, their trees look very different from each other even though they annotate the very similar information. Even worse is that it is not easy to compare each other nor to transform one to the other automatically. Therefore, it is necessary to transform a tree into a normalized form in order to compare syntactic annotations of different syntactic descriptions, and to extract valuable information from various formats of syntactic corpus in a uniform way.

In this paper, we devise a normalized description of a syntactic tree independent of methodologies and theories of syntactic analysis. The normalization uses a binary branching format of a rule. Hence, any rule format with a variable length of right hand side (RHS) can be normalized by rule binarization [4]. The normalization scheme is based on an automatic labelling algorithm using only POS tags to form a syntactic label of a construct. We use the term "construct" to mean a chunk, a constituent, or whatever being generated during tree construction. As we accept only POS tags for labelling syntactic categories without pre-defined non-terminals, trees constructed by different syntactic descriptions can be manipulated in the same manner given that they are based on the same POS tags. The syntactic label comes directly from POS tags so that it can reflect surface (i.e., POS tag) information onto the construct without discarding them. Each label is also of binary form like $DetNoun \rightarrow Det\ Noun$, where Det and $Noun$ come from each label of the two sub-constructs, determiner and noun. Each label is formed to represent both the state and the role* of the con-

Manuscript received January 23, 2003.
Manuscript revised May 6, 2003.
Final manuscript received June 13, 2003.

†Dept. of EECS, KAIST, Korea.
††Dept. of CSE, Ewha Womans Univ., Korea.

*Constructing a sentence from a sequence of morphemes can be seen as a state transition process. Then, every con-

struct in a sentence if possible. This scheme makes the resultant label keep sub-constructs' information as well as the label of the construct itself. The proposed rule normalization scheme with an automatic labelling algorithm is effective for the syntactic analysis of Korean. Moreover, this new scheme can support uniformity in manipulating any kinds of syntactic trees.

In the next section, we present characteristics of Korean and previous approaches to parsing Korean. After we propose our rule normalization with an automatic labelling scheme, a simple tool for normalization is introduced and a result of experiment on parsing efficiency is described. Finally, we discuss the applicability of the proposed method to other languages and conclude the work with future research areas.

2. Previous approaches to rule representation for parsing Korean

We will review the advantage and disadvantage of previous approaches in representing Korean syntax. Before we enter into the discussion of previous approaches in detail, we would like to consider the characteristics of Korean.

Korean is an agglutinative language. A sentence is made up of word phrases delimited by white spaces. A word phrase in turn is composed of one or more lexical morphemes concatenated by zero or more grammatical morphemes. Here, we need to choose word phrase or morpheme as the unit of rule description in order to describe a syntactic grammar for Korean. See the following sentence:

(1a) *sigan*/ncn†+*i*/jcs *gwijung*/ncps+*ha*/xsm+*n*/etm *geos*/nbn +*i*/jp +*da*/ef +./sf
(1b) time+*Subj* valuable+*Rel_Cls* thing+be+*Decl*+.
(1c) Time is a valuable thing.

Sentence (1) is made up of three word phrases, and nine morphemes including sentential mark. The infix operator '+' stands for the concatenation, agglutination in other words, of morphemes. Word phrase boundaries are marked by spaces. The underlined morpheme in the sentence (actually, '*i*/jp' is copula) should be concatenated to the noun phrase (NP) "*gwijunghan geos* (valuable thing)" only after that NP has been syntactically combined from the adnominal "*gwijunghan* (valuable)" and the noun "*geos* (thing)." Since Korean is a head-final language, the succeeding morpheme '*i*/jp' concatenated to the NP construct becomes a syntactic head, and the resultant construct acts as a predicate of the sentence. This characteristic, namely post-syntactic morpheme concatenation (PSMC), arises because the agglutination of the copula occurs after the two syntactic constructs were combined together to an NP.

Rules adopting word phrase as a unit of analysis encounter difficulties because of PSMC. Therefore, approaches that apply this kind of rules preprocess source corpus and input sentences to reconfigure the word phrases so as to make them cover PSMC, which in turn causes an undesirable source distortion [5], [6]. This limitation provides an opportunity to morpheme-based rules for the analysis of Korean.

Let us remind ourselves Fig. 1 in the previous section. The two syntactic structures shown in the figure are typical syntactic descriptions for sentence (1). The PSG structure in Fig. 1(a) is very similar to one in English in that the unit of syntactic annotation is word phrase, even though Korean is an agglutinative language where several morphemes (MPs) form a word phrase (WP) by agglutination.

Korean is a free word order language. Restricted form of phrase structure grammar (RPSG), as seen in Fig. 1(b), is proposed to cope with free word order characteristics of Korean [2], [7]. RPSG restricts grammar rules to three types: type 1 rules ($A \rightarrow B + \tau$) to describe derivation; type 2 rules ($A \rightarrow B + \gamma\ C$) to represent inter-constituent relations; and type 3 rules ($A \rightarrow A_1 + \gamma\ A_2 + \gamma\ \ldots\ A_n$) for coordination. It is implemented on morpheme-based syntactic unit.

Both PSG and RPSG confine syntactic labels to artificially predefined small sets (8 labels in [2], [7] and 11 labels in [1]). It results in the unintentional loss of intrinsic information melted in the POS tags, because the POS tags themselves reflect not only syntactic information but also morphotactic and semantic information [8].

The PSG structure doesn't properly describe the PSMC characteristics in the NP "*gwijung*+*ha*+*n* (valuable) *geos*(thing)" while RPSG covers it using type 2 rule "$NP \rightarrow ADJP + etm\ NP$". Besides, PSG doesn't transform NP to VP explicitly for "[*gwijung*+*ha*+*n*(valuable) *geos*(thing)]+*i*(be)" whereas RPSG applies type 1 rule "$VP \rightarrow NP + jp$" for the same purpose. On the other hand, PSG represents the subjective noun phrase "*sigan*(time)+*i*" as $NP - SBJ$ while RPSG provides no label for it. Since RPSG cares lexical morphemes as syntactic labels and treats grammatical morphemes only as a juncture between lexical morphemes, it can't represent the unit of scrambling of a sentence nor can give a thematic role therein.

struct at each state has a node label that defines the state of the construct at the point. The construct can also serve a role (i.e., a relationship) to other construct in the sentence. In this point of view, we regard a construct as having two different information - state and role. A 'state' indicates what the lexical delegate (i.e., content) of the construct is. A 'role' indicates the syntactic relationship (such as subject, object, etc.) of a construct to other construct.

†You can see the explanation of every symbol in Appendix A.

Table 1 Examples for each type of rules

	rule type	example	rule
1	$(C_L C_R \rightarrow f)$	*sigan*(time)/ncn	$0Ncn \rightarrow ncn$
2a	$(C_L C_R \rightarrow e + R_L R_R)$	*sil*(real)/xp + ($0Ncn$ *sigan*(time))	$1Ncn \rightarrow xp + 0Ncn$
2b	$(C_L C_R \rightarrow L_L L_R + e)$	($0Ncn$ *sigan*(time)) + *deul*(plural)/xsn	$1Ncn \rightarrow 0Ncn + xsn$
		($0Ncn$ *sigan*(time))+*i*(*Subj*)/jcs	$jcsNcn \rightarrow 0Ncn + jcs$
2c	$(C_L C_R \rightarrow L_L L_R + R_L R_R)$	($0Ncpa$ *geunmu*(duty)) + ($0Ncn$ *sigan*(time))	$NcpaNcn \rightarrow 0Ncpa + 0Ncn$
3	$(C_L C_R \rightarrow L_L L_R\ R_L R_R)$	($etmPaa$ *gwijunghan*(valuable)) ($0Nbn$ *geos*(thing))	$EtmNbn \rightarrow etmPaa\ 0Nbn$

3. Normalization of syntactic structures for Korean

In this section, we propose a new scheme to represent normalized Korean syntax with consideration of the advantages and disadvantage of previous approaches. Throughout this paper, we use the term "composite label" to point out the fact that the label C of a binary branching rule $C \rightarrow L\ R$ is composed from those of sub-construct L and of sub-construct R. In the following subsections, we will present the forms of binary branching rules and a method of automatic generation of composite labels for the rules.

3.1 Binary rules

Based on the reviews in the previous section, we adopt binary branching rules in this paper since binary branching rules are suitable for free word-order language. In designing a rule description scheme for Korean, we take care of the following considerations to be incorporated into the scheme:

- *No pre-defined syntactic labels instead of using POS tag information in labelling.* The syntactic labels are to be designed to use POS information. This means that we don't define a label independently of POS tags.
- *Reflection of Korean WP structures on rules.* In Korean, a grammatical morpheme can't appear by itself but always attaches to a construct by agglutination. Therefore, a grammatical morpheme doesn't need to be given an initial label, while a lexical morpheme is given a label. Rules also have to show agglutination and spacing by infix operators '+' and ' '.
- *PSMC.* The grammar should avoid a WP-based rule format. Each construct should be given a label. Each label of a construct should stand for both the state and the role of that construct.
- *Statistical invariance in probabilistic parsing.* Each candidate parse tree should comprise the same number of rules. It implies the length of RHS to be constant. It also achieves the typicality of a grammar.

A context-free grammar is a 4-tuple (T, N, G, S), where T and N are two finite disjoint sets of terminal (POS tag) and non-terminal (syntactic label) symbols, respectively, $S \in N$ is the start symbol, and G is a finite set of rules. In this paper, T is the KAIST tag set of 55 POS tags [8].

A rule is of binary form $C \rightarrow L\ R$ for the left hand side (LHS) label C to represent both dependency information and its syntactic property. L is a dependent and R is the head of the RHS in Korean since Korean is a head-final language. In addition, a lexical morpheme needs to be given a label while a grammatical morpheme doesn't. In order for a label C to represent the head of the RHS, the state of the construct (such as NP, VP in PSG), and the role of that construct (such as subject, object, etc.) at the same time, we need to devise a composite label of the form $C_L C_R$. Consequently, the rule $C \rightarrow L\ R$ is transformed to $C_L C_R \rightarrow L_L L_R\ R_L R_R$. With the fact that Korean has infix operators for agglutination ('+') and spacing (' '), we can define three rule types:

- Type 1 (unary rule): $C_L C_R \rightarrow f$, where $C_L C_R \in N, f \in T$ and f is a lexical morpheme.
- Type 2 (intra-WP rule): This type of rules covers all types of intra word-phrases. This type can be divided into three sub-types as follows:
  - 2a: $C_L C_R \rightarrow e + R_L R_R$;
  - 2b: $C_L C_R \rightarrow L_L L_R + e$; and
  - 2c: $C_L C_R \rightarrow L_L L_R + R_L R_R$,

  where $C_L C_R, L_L L_R, R_L R_R \in N, e \in T$, and e is a grammatical morpheme.
- Type 3 (inter-WP rule): $C_L C_R \rightarrow L_L L_R\ R_L R_R$, where $C_L C_R, L_L L_R, R_L R_R \in N$.

Table 1 shows examples for three types of rules, and their rules will be described in detail in the next subsection.

3.2 Automatic generation of composite labels

In this subsection, we describe how to generate a composite label for binary rules. To select proper composite labels, we need to observe the following constraints: any component labels should not be null; C_L and C_R may not take the same information to avoid duplication; and, the combination of component labels to make a composite label $C_L C_R$ should be in a uniform way. Both C_L and C_R are notated using morphemes themselves.

Table 2 Indicators for state and role

indicator	
state	$Ncpa, Ncps, Ncn, Nq, Nbu, Nbn, Npp, Npd, Nnc, Nno,$ $F, Ii, Pvd, Pvg, Pad, Paa, Px,$ $Mmd, Mma, Mad, Maj, Mag, Jp$
role	$jcs, jco, jcc, jcm, jcv, jca, jcj, jct, jcr, jxc, jxt, jxf,$ $ep, ecc, ecs, ecx, etn, etm, ef$

We denote L for the left sub-construct and R for the right sub-construct. $_{State}$ means the state tag of the construct. $_{role}$ is to represent the role of a grammatical morpheme in a construct, which will be applied to building a higher syntactic structure by combining with other construct. On the other hand, $_{Role}$ stands for the situation that the grammatical morpheme in the sub-construct has been applied to building the syntactic structure of the construct.

First, we define a state and a role of a construct using POS tags. We can explicitly map which morpheme indicates a state and which one indicates a role as in Table 2. States are the same as lexical morphemes except that 'Jp' is included therein. A state indicator starts with a upper case letter. Roles are in lower case letters. In other words, 'jcs' is a role indicator, but 'Jcs' is not.

When representing a composite label of a construct, we will use lower case letters for a role, but use a upper case letter for the first character of a state. For example, when a generated composite label of a construct is "$jcsNcn$," the state of the construct is 'Ncn', and the role is 'jcs'. However, a label "$EtmNbn$" has state 'Nbn' with no role because the word 'Etm' starting with upper case letter 'E' is not a role. Instead, this label means that a bound noun 'nbn' has been modified by an adnominal phrase "$etm*$."

Based on these notations, we can define a method of generating a composite label according to type of rules as follows:

- *Case 0: Initial case. a sub-construct has only one state.* Type 1, type 2a with a derivational prefix, and type 2b with a derivational suffix belong to this case. We use '0' for type 1 rules, and '1' for type 2 rules.
- *Case 1: L has only state and R has only role.* Type 2b (where e is not 'jp') belongs to this case. Composite label is $R_{role}L_{State}$, which means that role of the resultant construct would be R_{role} and the state is L_{State}.
- *Case 2: Both L and R have only states.* Type 2b (where e is 'jp') and type 2c are in this case. Composite label $L_{State}R_{State}$ means that the resultant construct has no role, but its state is R_{State}.
- *Case 3: L has a state and a role, R has only state.* Type 3 is in this case. The resultant construct has no role.
- *Case 4: L and R have both states and roles.* This is an exception of type 3.

Table 3 Composite labelling for each case

	Case 0 example	state	role	LHS label
L		-	-	$0R_{State}$
R	'$sigan$/ncn'	Ncn	-	$(0Ncn)$
	$0Ncn \rightarrow ncn$			(type 1)
L	'sil/xp'	-	-	$1R_{State}$
R	$+(0Ncn$ "$sigan$/ncn")	Ncn	-	$(1Ncn)$
	$1Ncn \rightarrow xp + 0Ncn$			(type 2a)
L	$(0Ncn$ "$sigan$/ncn")	Ncn	-	$1L_{State}$
R	'$+deul$/xsn'	-	-	$(1Ncn)$
	$1Ncn \rightarrow 0Ncn + xsn$			(type 2b)

	Case 1 example	state	role	LHS label
L	$(0Ncn$ "$sigan$/ncn")	Ncn	-	$R_{role}L_{State}$
R	'i/jcs'	-	jcs	$(jcsNcn)$
	$jcsNcn \rightarrow 0Ncn + jcs$			(type 2b)

	Case 2 example	state	role	LHS label
L	$(EtmNbn$ "$gwijunghan\ geos$")	Nbn	-	$L_{State}R_{State}$
R	'i/jp'	Jp	-	$(NbnJp)$
	$NbnJp \rightarrow EtmNbn + jp$			(type 2b)
L	$(0Ncpa$ "$geunmu$/ncpa")	$Ncpa$	-	$L_{State}R_{State}$
R	$+(0Ncn$ "$sigan$/ncn")	Ncn	-	$(NcpaNcn)$
	$NcpaNcn \rightarrow 0Ncpa + 0Ncn$			(type 2c)

	Case 3 example	state	role	LHS label
L	$(etmPaa$ "$gwijunghan$")	Paa	etm	$L_{role}R_{State}$
R	$(0Nbn$ "$geos$/nbn")	Nbn	-	$(EtmNbn)$
	$EtmNbn \rightarrow etmPaa\ 0Nbn$			(type 3)

	Case 4 example	state	role	LHS label
L	$(jcaNbn$ "$9sibuteo$")	Nbn	jca	$R_{role}R_{State}$
R	$(jcaNbn$ "$6siggaji$")	Nbn	jca	$(jcaNbn)$
	$jcaNbn \rightarrow jcaNbn\ jcaNbn$			(type 3)

Table 3 shows examples of composite labelling for each case. All the possible cases of combination of states and roles of a label for the binary rules will be 36 (because a label is composed of two elements and each of them can take 6 possible information $L_{State}, L_{Role}, L_{role}, R_{State}, R_{Role}$, and R_{role}). However, there are only five valid cases of combination as in the above. It's because (i) LL and RR compositions don't exist (case 4 is the only exception), (ii) role-to-role combinations without state information such as $L_{Role}R_{Role}$ are meaningless, (iii) a role component label always precedes a state component label in a composite label for the uniform representation, and (iv) $L_{role}R_{State}$ and $R_{Role}L_{State}$ can't represent a construct.

The explanation of the rule "$jcsNcn \rightarrow 0Ncn + jcs$" in case 1 for the construct "$sigan$(time)/ncn + $i(Subj)$/jcs" is as follows: First, because '$sigan$/ncn' is a lexical morpheme, the unary rule "$0Ncn \rightarrow ncn$" is applied to it; When it is combined to the following 'jcs,' the noun acts as a syntactic dependent; Since 'jcs' takes the role ($Subj$) of a construct, the composite la-

bel "*jcsNcn*" is formed to represent a subjective noun phrase. The lower case "*jcs*" in "*jcsNcn*" means that the subjective postposition follows the noun in word order. Fig. 2 shows the bracketed representation of sentence (1) using these composite labels.

In Fig.2, the rule "*JcsJp* → *jcsNcn NbnJp*" means that the subjective nominal construct "*jcsNcn*" is the dependent of a predicative head "*NbnJp*" so that the resultant construct "*JcsJp*" is a predicate with a subject.

In sum, the proposed grammar fulfills the considerations given in the beginning of this section. In addition, because it doesn't use any other information than the innate POS tag information of constructs, the grammar description scheme is open to other syntactic descriptions and to other languages as well.

4. Experiments

4.1 Introduction to a tool for normalizing syntactic structures

A Korean tree-tagging tool has been developed as an aid to help build a tree-annotated corpus of Korean. The tool based on a full parser performs normalization. It uses the KAIST POS tag set for annotating both morphotactic and syntactic information. As we have already discussed in Table 2, the indicators of state and role in POS tags are coordinated in the tool for executing normalization of syntactic descriptions. KAIST tree-tagged corpus is made up of 31,080 sentences with 55 POS tags and 8 syntactic tags in RPSG format [2], [7], [8]. A normalized corpus extracted from the KAIST corpus has the same POS tags and 726 composite labels. The normalization process in the tool can be seen in the following figures. An input syntactic tree of RPSG for sentence (1) is illustrated in Fig. 3. This shows the same tree information as that of Fig. 1(b). According to the normalization scheme described in section 3, the normalization process in itself

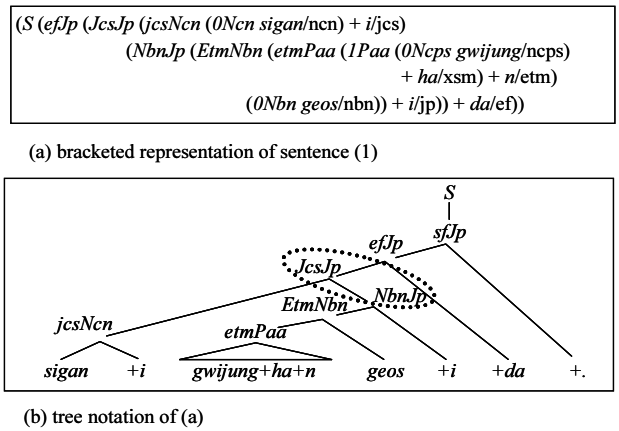


Fig. 2 Sentence (1) in the proposed normalization scheme

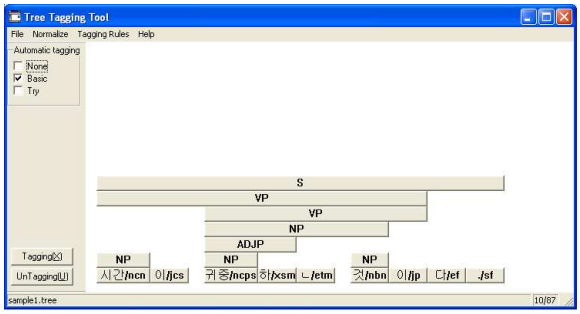


Fig. 3 Initial screen of RPSG syntactic tree for sentence (1)

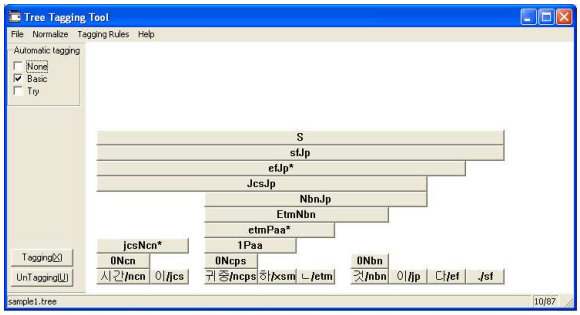


Fig. 4 Normalization of RPSG syntactic tree for sentence (1)

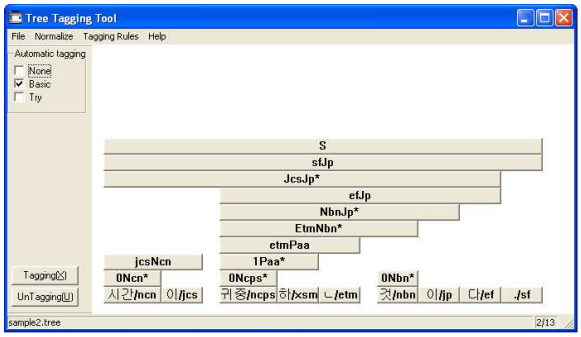


Fig. 5 Normalization of PSG syntactic tree for sentence (1)

generates binarized rules and their composite labels so that additional brackets and POS tag-based composite labels are created. Fig. 4 shows the resultant tree after normalization. It has all the same bracketing information as that of the original RPSG tree. The three differences are depicted by the asterisk in the figure, which means that the corresponding bracket of the label is automatically generated by the tool, not in the original tree.

In case of Penn Korean Treebank, it consists of around 54,000 words and 5,000 sentences and uses a phrase structure style of annotation, making head/phrasal node distinctions, argument/adjunct distinctions, and identifying empty arguments and traces for moved constituents. This uses 34 POS tags and 11 syntactic symbols of combination of clause level and phrase level. Since Penn Korean Treebank uses word

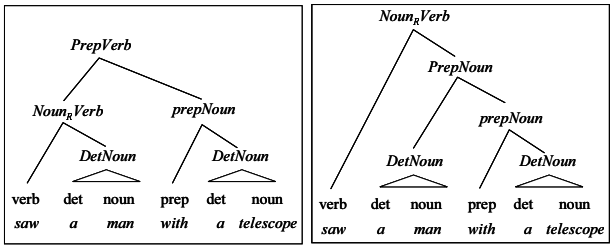


Fig. 6 Applying normalization scheme to a PP in English

phrase-based bracketing, the tool adds systematically an additional bracket for each morpheme, which is not in the original input, from left-to-right within the span of an original bracket in order to make the output the corresponding, morpheme-based tree.

After mapping Penn Korean POS tag set into corresponding KAIST one, the tool generates a normalized tree for the PSG tree of sentence (1) as depicted in Fig. 5. In this figure, seven labels and their brackets are generated automatically by the tool.

In Fig. 4 and 5, we can see that the normalized PSG and RPSG trees are the same except that "$efJp$" and "$JcsJp$" in PSG don't match with those in RPSG. "$efJp$" and "$JcsJp$" are included in both Fig. 4 and Fig. 5, but their spans in each figure are different. This is because the PSG annotation does not support PSMC characteristic of Korean. Through the normalization, we can see that it is easy to compare different syntactic descriptions without difficulty given that the innate POS tags adopted are the same.

4.2 Evaluation of the proposed normalization scheme

Some approaches analyze alternative tree representations with an estimation method of relative frequency, thereby proposing that parent annotation is a better way of encoding syntactic contexts in rules [9], [10]. For the evaluation of the proposed representation scheme, a typical example of prepositional phrase(PP) attachment in English is given in Fig. 6.†

Using the estimation method given in [9], the relative frequency f and the estimated frequency $\hat{f}$ become quite the same.†† It means that the proposed scheme fairly encodes syntactic contexts in rules. The differences come from (1) the annotation: [9] adopts parent annotation in node labelling, while we adopt descendants annotation; and (2) syntactic tag set: [9] uses a limited number of artificially defined syntactic tags (i.e., Penn syntactic tags), whereas we use POS tags.

Furthermore, we compare a new normalization scheme with an existing syntactic description in order to prove that this scheme does not lose systematic efficiency. We transform the KAIST corpus into that of proposed rule format and divide it into 10 parts for context-free cross validation (CV). The input to the parser is the sequence of POS tags. For evaluation, we adopt EVALB program [11] that applies PARSEVAL measure [12]. Table 4 depicts the test results of grammars on the KAIST corpus.

Table 4 Test results of the grammars on the KAIST corpus

syntactic desc	test set	labels	rules	ambiguity	LP	LR
RPSG	31,080 CV	8	2,381	3.76 e^{14}	75.39	72.83
proposed	31,080 CV	726	10,735	4.69 e^{6}	75.58	75.58

The first experiment is cross validation result of RPSG in the same environment as in [2]. RPSG uses 8 syntactic labels (S, NP, VP, $ADJP$, $ADVP$, $MODP$, $AUXP$, IP). Its result is around 75% accuracy for Korean, which conforms to the presupposition in English such as probabilistic context-free parsing for English yields around 75% accuracy for precision and recall [13]. (For example, the labelled precision (LP) and the labelled recall (LR) for Wall Street Journal section 23 of the Penn Treebank are 74.3% and 70.1%, respectively [14].)

The second experiment has been run using the proposed rule description scheme. The number of automatically extracted labels of the proposed grammar is 726. As we can denote a label by $C_L C_R$, and C_R always represents a state, there can be 31 candidate tags for C_R (symbol, nominal, foreign, interjection, predicate, modifier, and copula of the Table A.1). 68 candidate tags for C_L can also be estimated in the same way. Among the possible 2,108 labels, there are morphotactic and syntactic connectivity restrictions in combining morphemes or constructs so that only 726 labels are found in the corpus.

The criterion of the evaluation of an output tree is whether the labels of the nodes in the tree are identical to those in the gold parse tree. That induces a fact that the accuracy of a parser would be low if the parser employs a set of non-terminals with big cardinality. The number of composite labels in the second experiment is still too big to get a higher result, even though the accuracy is slightly better than that of RPSG. However, since all the composite non-terminals such as "$EtmNbn$" can be clustered to some set of non-composite non-terminals such as "Ncn," we can cluster 726 composite labels into 55 categories using a simple heuristic, resulting in 79.30% accuracy [15].

†We arbitrarily define POS tags of English for the sake of explanation. '$prep$' in "$prepNoun$" means the preposition's role is not yet applied to form a higher construct, whereas '$Prep$' in "$PrepNoun$" means that the role has been applied. '$Noun_R$' is to make explicit the word order so that we can know that the nominal came from right subconstruct.

††For $f = 0.48$, the estimated relative frequency $\hat{f} = 0.48$ whereas $\hat{f} = 0.46$ in [9].

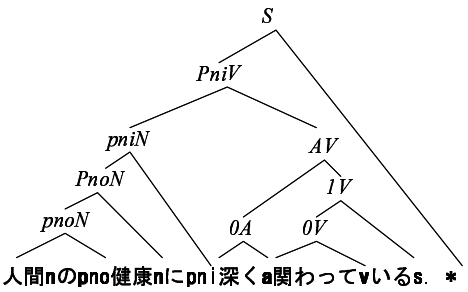


Fig. 7 Applying normalization scheme to Japanese

5. Discussion

We discuss the applicability of the proposed approach to other languages, such as Japanese and English in this section. Japanese is also an agglutinative language and has similarities with Korean, so we expect that we can apply this scheme to Japanese with little effort. Fig. 7 shows the application of the proposed scheme on a Japanese sentence, which means "it is deeply related to the health of human."†††

The particles attached to nouns and the suffixes (or endings†) attached to the verb take charge of the roles in the sentence. Although it might get complicated in applying this scheme to Japanese in bulk, Japanese shares the benefits of this scheme with a minor modification.

In case of inflectional languages such as English, the position of a word in a sentence decides the syntactic role of the word in most cases, even though some POS tags - preposition, relative pronoun, conjunction, etc. - have role information by themselves. Fig. 6 depicts a simple example of applying this scheme to a prepositional phrase attachment of English. Adapting this scheme to inflectional languages may invoke changes to the scheme, but how to handle role information connoted in a word order is left for future study.

6. Conclusion

We analyzed Korean syntactic characteristics and existing grammars. The proposed normalization scheme and a syntactic annotation adopt binary branching rules with composite labels automatically generated using only POS tags. The resultant rule scheme naturally represents both the state and the role of a construct, and both the dependency and the syntactic information of that construct during the parsing. It also meets the PSMC characteristic and avoids a statistical bias.

The scheme uses only POS information in the input so that we can easily get normalized syntactic structures from different syntactic descriptions given that their POS information are the same. In addition, because it doesn't use any other information than the innate POS tag information of constructs, the rule description scheme is open to other languages as well.

From a performance point of view, this scheme shows 75.58% LP and LR, which are higher results than those of RPSG. The current rule description scheme doesn't use contextual information such as head-to-head collocation, outside context, distance information, or argument structures at all. Those information can readily be melted in the scheme to yield higher performance.

Appendix A. KAIST POS Tag Set

Table A.1. describes the KAIST POS tag set.

Acknowledgement

This research was supported by the Ministry of Science and Technology in Korea.

References

[1] C. H. Han, N. R. Han, and E. S. Ko, "Bracketing Guidelines for Penn Korean TreeBank," IRCS Report 01-10, University of Pennsylvania, 2001.
[2] K. J. Lee, J. H. Kim, and G. C. Kim, "An efficient parsing of Korean sentences using restricted phrase structure grammar," Computer Processing of Oriental Languages 11(1), 1997, 49 - 62.
[3] The KAIST corpus 1996-1997, Korea Advanced Institute of Science and Tehcnology, 1997, http://korterm.org/.
[4] C. Manning and B. Carpenter, "Probabilistic parsing using left corner language models," Proceedings of the Fifth International Workshop on Parsing Technologies (IWPT-97), MIT, 1994, 147-158.
[5] C. H. Kim, J. H. Kim, J. Y. Seo, and G. C. Kim, "A right-to-left chart parsing with headable paths for Korean dependency grammar," Computer Processing of Chinese and Oriental Languages 8 (Supplement), 1994, 105-118.
[6] K. J. Seo, K. C. Nam, and K. S. Choi, "A probabilistic model for dependency parsing considering ascending dependencies," Literary and Linguistic Computing 13(2), 1998, 59-63.
[7] K. J. Lee, Probabilistic Parsing of Korean Based on Language-Specific Properties, Ph.D. Thesis, Department of Computer Science, Korea Advanced Institute of Science and Technology, Taejon, 1998.
[8] K. S. Choi, Y. J. Nam, J. G. Kim, Y. G. Han, S. M. Park, J. S. Kim, C. T. Lee, D. B. Kim, J. H. Kim, and B. J. Choi, "A study of the morphological and syntactic tag standard for Korean language information bases," Korean Journal of Cognitive Science 7(4), 1996, 43-61.
[9] M. Johnson, "The effect of alternative tree representations on tree bank grammars," D.M.W. Powers (ed.),

†††POS information is shown at the tail of each word: n(noun), p(particle), a(adverb), v(verb), s(suffix), and *(symbol) [16]. We break down the particle(p) into *pni* and *pno* in this example for discriminating the composite labels.

†We are not sure whether endings are separately analyzed from verb/adjective in case of Japanese. However, it is good to tag them separately for using this scheme.

Table A· 1 The KAIST POS tag set

morpheme		tag
lexical	**S**ymbol	sp (pause), sf (full stop), sl (left quotation & parenthesis mark), sr (right quotation & parenthesis mark), sd (dash), se (ellipsis), su (unit), sy (other symbols)
morpheme	**N**ominal	ncpa (active-predicative common noun), ncps (stative-predicative common noun), ncn (non-predicative common noun), nq (proper noun), nbu (unit bound noun), nbn (non-unit bound noun), npp (personal pronoun), npd (demonstrative pronoun), nnc (cardinal numerals), nno (ordinal numerals)
	Foreign	f (foreign word)
	Intj	ii (interjection)
	Predicate	pvd (demonstrative verb), pvg (general verb), pad (demonstrative adjective), paa (attributive adjective), px (auxiliary verb)
	Modifier	mmd (demonstrative adnoun), mma (attributive adnoun), mad (demonstrative adverb), maj (conjunctive adverb), mag (general adverb)
grammatical	**J**osa	jcs (subjective), jco (objective), jcc (complemental), jcm (adnominal), jcv (vocative), jca (adverbial), jcj (conjunctive), jct (comitative), jcr (quotative), jxc (common auxiliary), jxt (topical auxiliary), jxf (final auxiliary), jp (predicative case)
morpheme	**E**nding	ep (prefinal), ecc (coordinate), ecs (subordinate), ecx (auxiliary conjunctive), etn (nominalizing), etm (adnominalizing), ef (final)
	Affix	xp (prefix), xsn (noun-derivational), xsv (verb-derivational), xsm (adjective-derivational), xsa (adverb-derivational)

NeMLaP3/CoNLL98: New Methods in Language Processing and Computational Natural Language Learning, ACL, 1998, 39-48.

[10] M. Johnson, "PCFG models of linguistic tree representations," Computational Linguistics 24(4), 1998, 613-632.

[11] S. Sekine and M. Collins, Evalb, 1997, ftp://cs.nyu.edu/.

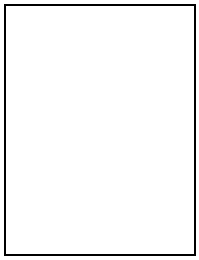
[12] E. Black, S. Abney, D. Flickinger, C. Gdaniec, R. Grishman, P. Harrison, D. Hindle, R. Ingria, F. Jelinek, J. Klavans, M. Liberman, M. Marcus, S. Roukos, B. Santorini, and T. Strzalkowski, "A procedure for quantitatively comparing the syntactic coverage of English grammars," Proceedings of Speech and Natural Language Workshop, DARPA, Pacific Grove, 1991, 306-311.

[13] C. M. White, Rapid Grammar Development and Parsing Constraint Dependency Grammars with Abstract Role Values, Ph.D. Thesis, Purdue University, 2000.

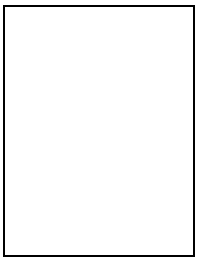
[14] E. Charniak, "Statistical parsing with a context-free grammar and word statistics," AAAI, 1997, 598-603.

[15] S. Y. Kim, K. J. Lee, and K. S. Choi, "Automatic generation of composite labels using part-of-speech tags for parsing Korean," International Journal of Computer Processing on Oriental Languages 16(2), 2003.

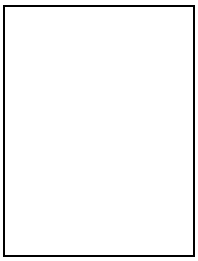
[16] http://www-nagao.kuee.kyoto-u.ac.jp/.



Seongyong Kim received the B.S. degree in computer science and statistics from Seoul National University, Korea, in 1985, the M.S. degree in computer science from Korea Advanced Institute of Science and Technology (KAIST) in 1987. He is currently working toward the Ph.D. degree at the Division of Computer Science, Department of Electrical Engineering and Computer Science, KAIST. Since 1987, he has been working for Agency for Defense Development in Korea. His research interests include natural language parsing, information retrieval, and intelligent agents.



Kong Joo Lee received the B.S. degree in computer science from the Sogang University, Korea, in 1992, the M.S. degree in 1994 and the Ph.D. degree in 1998 in computer science from KAIST. During 1998 through 2002 she was working for Microsoft, Korea. She is currently a faculty member of the department of computer science in Ewha Womans University. Her research interests include information retrieval, natural language parsing, and machine translation.



Key-Sun Choi received the B.S. degree in mathematics in Seoul National University, Korea, in 1978, the M.S. degree in 1980 and the Ph.D. degree in 1986 in computer science from KAIST. From 1985 to 1986, he was a professor in Hangook University for Foreign Studies. In 1987 and 1988, he was an invited researcher in NEC C&C, Japan. Since 1988, he has been a faculty member of the Department of Computer Science and also a director of KORTERM, KAIST. His research interests include natural language processing, machine translation, information retrieval, and terminology.