Customizing the XTAG System for Efficient Grammar Development for Korean[*]

Juntae Yoon[†], Chung-hye Han[†], Nari Kim[‡] and Meesook Kim[†]

[†]IRCS, University of Pennsylvania
3401 Walnut St., Suite 400
Philadelphia, PA 19104
USA

[‡]Konan Technology, Inc.
Simone Building, 144-1
Samsung-Dong, Kangnam-Gu
Seoul 135-090, Korea

{jtyoon,chunghye,nari,meesook}@linc.cis.upenn.edu

Abstract

This paper addresses linguistic and implementation problems for a practical LTAG parser raised by rich morphology in Korean. We propose a way of representing the Korean inflectional system as feature structures in lexicalized elementary trees, and describe our implemented modifications on the XTAG system for a more efficient grammar development for Korean.

1. Issues

Korean is an agglutinative language with a very productive inflectional system. Inflections include postpositions on nouns; tense morphemes and endings that indicate sentence types on verbs and adjectives; among others. Furthermore, these inflections can combine with each other to form compound inflections.

(1) Noun

a. *hakkyo-ka*
school-Nom

b. *hakkyo-eyse-ka*
school-from-Nom

c. *hakkyo-eyse-man*
school-from-only

d. *hakkyo-eyse-man-un*
school-from-only-Topic

(2) Verb

a. *ka-ss-ta*
go-Past-Decl

b. *ka-si-ess–ta*
go-Honor-Past-Decl

c. *ka-ki-ka*
go-Nominalizer-Nom

d. *ka-si-ess-ki-ey-nun*
go-Honor-Past-Nominalizer-to-Topic

This implies that a word in Korean can have a very large number of morphological variants. For example, verbs can be followed by honorific and tense morphemes which can then be followed by endings indicating clause-type which then can be followed by case postpositions. Similarly, adverbial postpositions which correspond to English prepositions, can be followed by other case postpositions such as nominative or accusative case markers and auxiliary postpositions such as *to* (‘also’) and *man* (‘only’), which then can be followed by a topic marker. Accordingly,

[*] We thank Martha Palmer, Aravind Joshi, Anoop Sarkar and the XTAG Group at Penn for their support and discussion. We also acknowledge the two anonymous reviewers. This project has been partially funded by the Army Research Lab via a subcontract from CoGenTex, Inc. and NSF Grant SBR 8920230. The third author’s contribution to this work was made while she was a visiting researcher at IRCS.

the number of possible morphological variants of a word can in principle be in the tens of thousands.

This property of Korean raises two issues within the context of developing and implementing a Feature Based Lexicalized Tree Adjoining Grammar (FB-LTAG) for Korean using the XTAG system (The XTAG-Group, 1998): (1) adequate linguistic description of the inflections and (2) efficient lexicon development. From a linguistic point of view, describing a grammar of a language is to construct rules that generate sentences in the language at a formal level. From an implementational point of view, the grammar should be described in a consistent and efficient way. The XTAG system helps us to pursue both these goals, but the complicated inflection system mentioned above leads to difficulties in building a grammar for Korean.

In this paper, we provide our solution to the linguistic and implementational issues raised by these morphological properties of Korean. We first provide a way of handling the Korean inflectional system using feature structures in lexicalized elementary trees in section 2. We impose a hierarchy on various types of inflections in order to handle all possible ways of combining inflections, and we represent this by assigning different feature attributes to different types of inflections.

In section 3, we then point out that the current XTAG system as it is forces us to construct a lexicon (i.e., syntactic database) that lists all possible morphological variants of words. A lexicon must contain all possible *eojeols*, where an *eojeol* is a term in Korean for denoting a spacing unit which consists of a content word and associated functional words. However, this is highly impractical and inefficient given the rich inflectional system in Korean. We would end up with a very large (even unbounded) lexicon. Therefore, we found it necessary to develop an alternative method for constructing the lexicon in order to continue to use the XTAG parser for developing a Korean grammar . One possible solution to the problem is to incorporate morphological rules in the grammar that regulate the generation of *eojeols* with several morphemes combined. However, doing so will mix up morphological generative rules with syntactic rules, complicating the TAG grammar tremendously. Instead, we have chosen to pursue an approach in which morphological regularities are handled by a separate morphological component using a morphological analyzer (Yoon *et al.*, 1999). The output of this analysis then interacts with our Korean TAG grammar which handles syntactic regularities. As a way of implementing this approach, we modified the XTAG system by dividing up the syntactic database into elementary syntactic database (ESDB) and local syntactic database (LSDB). ESDB is a general lexicon that contains stems with the elementary trees associated with them. LSDB is a partial lexicon dynamically generated for each input sentence using information from ESDB and the output of a morphological analyzer. That is, it contains only entries for *eojeols* occurring in the input sentence. The morphological analyzer produces the morphological analysis of each *eojeol* in the input sentence identifying its stem and inflections. Then, the stem of each *eojeol* is associated with elementary trees or tree families by looking up the ESDB and stored in the LSDB. The inflections of each *eojeol* are converted into features and are also stored in the LSDB. This modification to the XTAG system allows us to build a lexicon efficiently and develop a grammar for Korean that is compatible with the XTAG system.

2. Handling inflectional morphology

In our current Korean grammar, the inflectional morphology on an *eojeol* that are relevant for syntactic analysis is represented as features on the tree node. For instance, a noun with a nominative case marker is associated with the feature <case:nom> and when this lexical item is anchored by an NP tree, the feature <case:nom> is passed up to the NP node.

In Korean, combining inflections is a highly productive process with some restrictions. For example, nominative, accusative and genitive CASE postpositions occur in a complementary distribution, but ADVERBIAL postpositions (which correspond to English prepositions) such as -*ey* (‘at’), *eykey* (‘from’), -*kkaci* (‘to’), etc. can be followed by nominative case or genitive case. Case and adverbial postpositions are assumed to be assigned by the predicate of the sentence. Moreover, AUXILIARY postpositions which have semantic content such as -*man* (‘only’) and -*to* (‘even’) can combine with an adverbial postposition and the topic marker -*(n)un* can combine with an adverbial postposition and/or an auxiliary postposition but not the case postposition.

Moreover, predicates[1] in Korean are inflected with several morphemes. They carry CLAUSE-TYPE morphemes that indicate whether the clause is a main, coordinate, subordinate, relative clause, or nominalized clause. If a clause is a main clause, the verb carries a MODE morpheme that indicates whether the clause is a declarative, imperative, interrogative, exclamation, or propositive, etc. Clause-type morphemes and mode morphemes occur at the end of the verb. In addition, verbs also carry TENSE inflections right before the clause-type and mode morphemes. Further, all these inflections can be expressed in many different ways.

In order to handle all possible ways of combining inflections, we imposed a hierarchy among various types of inflections and represented this by assigning different types of inflections to different feature attributes. Table 1 summarizes the list of inflectional feature attributes and the corresponding feature values currently being used by our grammar. The label ‘pp’ on <adv-pp> and <aux-pp> stand for postpositions. Note that verbal features include <ending> which allows us to store the string values of mode and clause-type morphemes in the tree node for later semantic interpretation. Examples of an NP tree that anchors a noun (*hakkyo* ‘school’) with compound inflections, and an S tree that anchors a verb (*ka* ‘go’) with some verbal inflections are given in Figure 1.

On nouns		
⟨**case**⟩	a case feature assigned by predicate	nom, acc, gen, adv
<**adv-pp**>	a feature assigned by predicate only if <case:adv>, which corresponds to English prepositions such as *to, from, in*	string values such as *ey, eyse, lo, wa, ya, kkaci, pwute, pota, lako, losse, ...*
<**topic**>	presence/absence of topic marker	+, -
<**aux-pp**>	adds specific meaning e.g., *only, also*	string values e.g., *to, man*
On predicates		
<**clause-type**>	a feature that indicates the type of the clause that contains the predicate	main, coord, subord, adnom, nominal, aux-connect
<**mode**>	a feature on a predicate only if <clause-type:main>	decl, imp, int, excl, propos
<**tense**>	encodes temporal interpretation	pres, past, future
<**ending**>	a feature marked for different ways of instantiating mode and the clausal type	string values e.g., *ta, nunka, ela, ki, nun, tako, ...*

Table 1: Features for Inflectional Morphology

3. Local Syntactic Database

Our Korean XTAG system uses the LTAG parser developed by Anoop Sarkar (Sarkar, 2000). Written in C, it can process Korean characters represented as 2-byte codes. This parser was meant to use the XTAG English grammar (The XTAG-Group, 1998), and so it uses the lexical

[1] In Korean, both verbs and adjectives play the role of a predicate in a sentence.

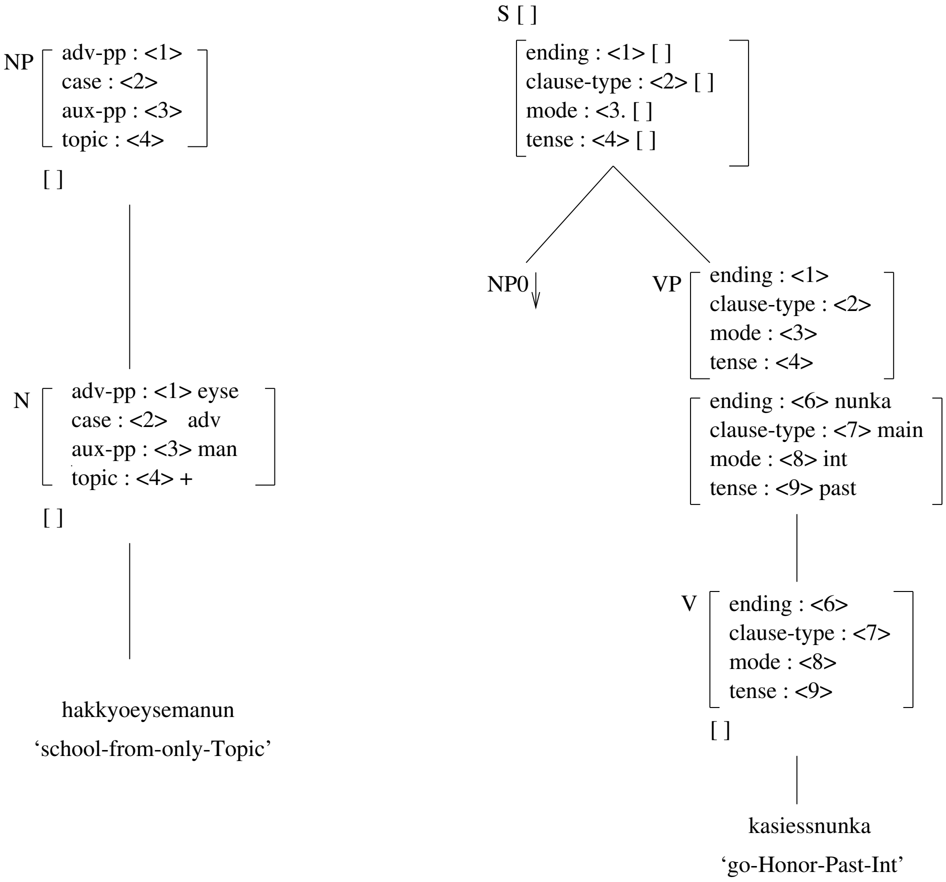

Figure 1: Instantiating Inflections as Features

databases that are part of the English grammar. In the English grammar, all the morphological variants of each word are listed in the morphological database (Morph DB), where they are mapped to a stem and lexical feature structures. The stem is then used to select a set of elementary trees in the syntactic database. The older Common Lisp XTAG parser keeps these databases separate, but the C parser combines them into a single database. The C parser uses this database (Syn DB) in order to select appropriate trees for the words in the input sentence.

Since a word in English has a small number of inflections, it is possible to describe as separate entries all the inflected forms in the Syn DB. However, this way of describing lexicons for Korean is impractical and inefficient, due to its rich morphology. To resolve this problem, we separate the Syn DB into Elementary Syn DB (ESDB) and Local Syn DB (LSDB). Only stems of *eojeols* are listed in the ESDB. This means that we can construct a lexicon for the stem words without considering all the morphological variants, making the life of grammar developers much easier. LSDB only contains *eojeols* of an input sentence as entries with associated elementary trees and lexical feature structures. LSDB is generated dynamically through the use of *Lexicon Extractor*. Given an input sentence, the lexicon extractor takes the result of morphological analysis produced by a morphological analyzer (and the POS tagger) developed at Yonsei University (Yoon *et al.*, 1999), and generates an LSDB by making reference to the ESDB and converting inflections on each *eojeol* to feature structures.

The step of generating an LSDB is as follows:

Firstly, the input sentence goes through the morphological analyzer and the POS tagger. If the morphological analyzer or the POS tagger makes errors, the user can manually input the tagged

sotaycangi	*sotaycang*/N+*i*/Pn
mwucenkilul	*mwucenki*/N+*lul*/Pa
swulihayessta	*swuliha*/V+*yess*/PE+*ta*/Ei+./SC

Table 2: Results of the morphological analyzer and the POS tagger for example (3)

form. We consider the example sentence in (3) to show the execution of the system.

(3) *Sotaycang-i mwucenki-lul swuliha-yess-ta.*
platoon-leader-Nom radio-Acc repair-Past-Decl
'The platoon leader repaired the radio.'

The morphological and tagging results of (3) are as shown in Table 2. Here, N, V, PE, Ei, Pn, Pa, and SC are tags for noun, verb, pre-ending (i.e., tense), indicative ending, nominative postposition, accusative postposition and period punctuation respectively.[2]

Secondly, the lexicon extractor extracts syntactic information from ESDB for the content words and function words which appear in the given sentence, i.e. 'sotaycang/N', 'i/Pn', 'mwucenki/N', 'lul/Pa', 'swuliha/V', 'yess/PE', 'ta/Ei' and './SC'. From the results of morphological analysis, the lexicon extractor selects elementary trees or tree families for the content words (i.e., 'sotaycang/N', 'mwucenki/N', 'swuliha/V') by looking up the ESDB. The function words (i.e., 'i/Pn', 'lul/Pa', 'yess/PE' and 'ta/Ei') are converted to feature structures, which will appear in tree nodes. With this data collected, the LSDB is generated listing all the *eojeols* in the input sentence with associated elementary trees, tree families and features. Crucially, the LSDB contains only the *eojeols* of the input sentence as entries. Table 3 shows the LSDB generated from the morphological analysis results and the ESDB. In Table 3, @nom is a template for <case>=nom, @acc for <case>=acc, @past for <tense>=past, @cls-main for <clause-type>=main and @end-*ta* for <ending>=ta.

⟨⟨INDEX⟩⟩*sotaycangi* ⟨⟨ENTRY⟩⟩*sotaycangi* ⟨⟨POS⟩⟩N ⟨⟨TREES⟩⟩αNP βNP βNP-V βNP-S ⟨⟨FEATURES⟩⟩@nom
⟨⟨INDEX⟩⟩*mwucenkilul* ⟨⟨ENTRY⟩⟩*mwucenkilul* ⟨⟨POS⟩⟩N ⟨⟨TREES⟩⟩αNP βNP βNP-V βNP-S ⟨⟨FEATURES⟩⟩@acc
⟨⟨INDEX⟩⟩*swulihayessta* ⟨⟨ENTRY⟩⟩*swulihayessta* ⟨⟨POS⟩⟩V ⟨⟨FAMILY⟩⟩Tnx0nx1V ⟨⟨FEATURES⟩⟩@past @cls-main @end-*ta*

Table 3: LSDB generated from the example

The parser uses this LSDB and generates the derived tree shown in Figure 2.

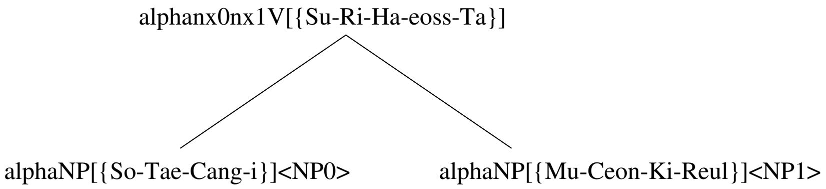
alphanx0nx1V[{Su-Ri-Ha-eoss-Ta}]
alphaNP[{So-Tae-Cang-i}]<NP0>
alphaNP[{Mu-Ceon-Ki-Reul}]<NP1>

Figure 2: Derivation tree for the example sentence (3)

[2] Although our system reads and generates Hangul (Korean characters), we use romanized examples in this paper for convenience.

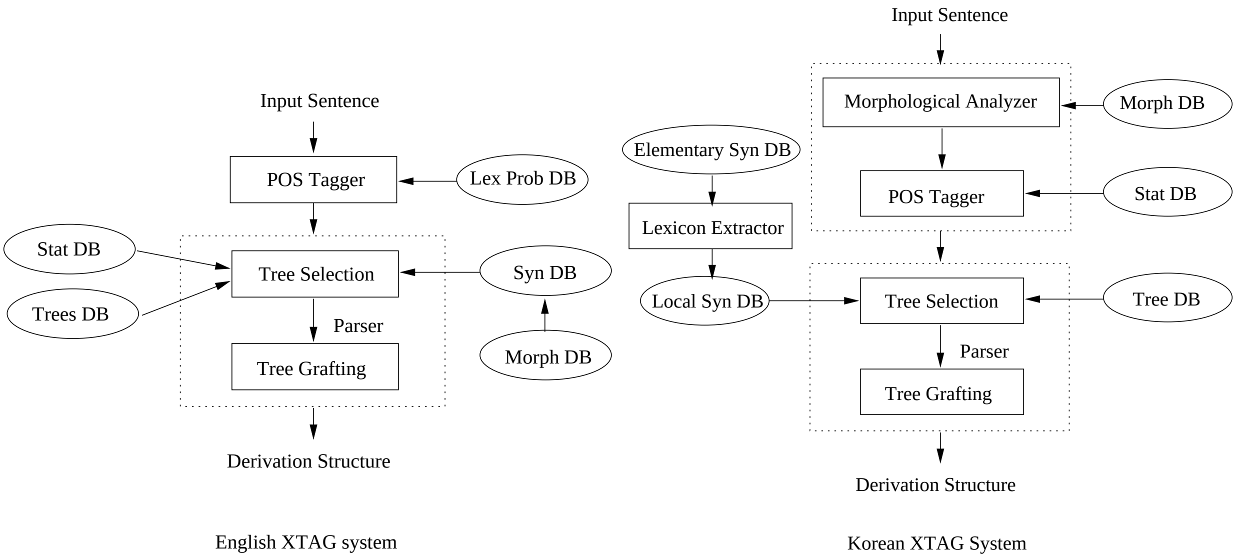

Figure 3: Korean XTAG System and English XTAG System

The overall flow of our Korean XTAG system is represented in Figure 3 in comparison to the English XTAG system. In the Korean XTAG system, we added the lexicon extractor, morphological analyzer and POS tagger which run independently of the XTAG parser. Our Korean grammar currently has 15 tree families and 289 elementary trees that handle various syntactic phenomena: e.g., adverb modification, sentences with empty arguments, relative clauses, complex noun phrases, auxiliary verbs, gerunds, adjunct clauses.

4. Conclusion

In this paper, we addressed linguistic and implementation problems raised by rich morphology in Korean. We first motivated a feature hierarchy on various types of inflections in order to handle all possible ways of combing them. We then described the modifications we have implemented on the English XTAG system, enriching it with a morphological analyzer (which also does POS tagging) and lexicon extractor. These modifications enable us to get rid of a syntactic database from the system that would require listing of all possible morphological variants of words. Instead, we divide up the syntactic database into ESDB and LSDB, where ESDB contains stems with associated elementary trees and tree families, and LSDB only contains *eojeols* of a given input sentence with associated elementary trees and feature structures to represent inflections. Furthermore, by incorporating a morphological analyzer to the system, we are able to separate out morphological generative rules from syntactic rules in the description of LTAG grammar for Korean. Our approach can be applied to FB-LTAG development for other languages with rich morphology.

References

SARKAR A. (2000). A probabilistic head-corner chart parser for TAGs. Ms. UPenn.

THE XTAG-GROUP (1998). *A Lexicalized Tree Adjoining Grammar for English*. Technical Report IRCS 98-18, UPenn.

YOON J., LEE C., KIM S. & SONG M. (1999). Morphological analyzer of Yonsei univ., Morany: Morphological analysis based on large lexical database extracted from corpus. In *Proceedings of Korean Language Information Processing* (In Korean).