

# Penn Korean Treebank: Development and Evaluation

Chung-hye Han  
Dept. of Linguistics  
Simon Fraser University  
8888 University Drive  
Burnaby BC V5A 1S6, Canada  
chunghye@sfu.ca

**Martha Palmer**  
Dept. of Computer Information and Science  
University of Pennsylvania  
256 Moore School  
Philadelphia, PA 19104, USA  
mpalmer@linc.cis.upenn.edu  
Na-Rae Han, Eon-Suk Ko  
Dept. of Linguistics  
University of Pennsylvania  
619 Williams Hall  
Philadelphia, PA 19104, USA  
{nrh,esko}@ling.upenn.edu

**Heejong Yi**  
Dept. of Linguistics  
University of Delaware  
46E. Delaware Ave.  
Newark, DE 19716, USA  
hyi@udel.edu

December 10, 2001

## Abstract

This paper discusses issues in building a 54-thousand-word Korean Treebank using a phrase structure annotation, along with developing annotation guidelines based on the morpho-syntactic phenomena represented in the corpus. Various methods that were employed for quality control and the evaluation on the Treebank are also presented.

## 1 Introduction

With growing interest in Korean language processing, numerous natural languages processing (NLP) tools for Korean, such as part-of-speech (POS) taggers, morphological analyzers, parsers, have been developed. This progress was possible through the availability of large-scale raw text corpora and POS tagged corpora (etri99,yoon-corpus99-2,yoon-corpus99-1). However, no large-scale bracketed corpora are currently available to the public, although efforts have been made to develop guidelines for syntactic annotation (kaist-tb1,kaist-tb2). As a first step towards addressing this issue, we have built a 54-thousand-word<sup>1</sup> Korean Treebank using a phrase structure annotation, along with developing annotation guidelines based on the morpho-syntactic phenomena represented in the corpus, over the period of Jan. 2000 and April 2001. The corpus that we used for the Korean Treebank consists of texts from military language training manuals. These texts contain information about various aspects of the military, such as troop movement, intelligence gathering, and equipment supplies, among others. This corpus is part of a Korean/English bilingual corpora that was used for domain specific Korean/English machine translation project at the University of Pennsylvania. One of the main reasons for annotating this corpus was to train taggers and parsers that can be used for the MT project.

In this paper, we first discuss some issues in developing the annotation guidelines for POS tagging and syntactic bracketing. We then detail the annotation process in §??, including various methods we used to detect and correct annotation errors. §?? presents some statistics on the size of the corpus, and §?? discusses the results of the evaluation on the Treebank.

## 2 Guideline development

The guiding principles employed in developing the annotation guidelines were theory-neutralness (whenever possible), descriptive accuracy and con-

---

<sup>1</sup>This word count is computed on tokenized texts and includes symbols.

sistency. To this end, various existing knowledge sources were consulted, including theoretical linguistic literature on Korean, publications on Korean descriptive grammar, as well as research works on building tagged Korean corpora by such institutions as KAIST and ETRI (etri99,kaist-tb1,kaist-tb2,yoon-corpus99-2,yoon-corpus99-1). Ideally, complete guidelines should be available to the annotators before annotation begins. However, linguistic problems posed by corpus is much more diverse and complicated than those discussed in theoretical linguistics or grammar books, and new problems surface as we annotate more data. Hence, our guidelines were revised, updated and enriched incrementally as the annotation process progressed. In cases where no agreement could be reached among several alternatives, the one most consistent with the overall guidelines was chosen, with the consideration that the annotated corpus may be converted to accommodate other alternatives when needed. In the next two subsections, we describe in more detail the main points of POS tagging guidelines and syntactic bracketing guidelines.

## 2.1 POS tagging and morphological analysis

Korean is an agglutinative language with a very productive inflectional system. Inflections include postpositions, suffixes and prefixes on nouns, and tense morphemes, honorifics and other endings on verbs and adjectives. For this reason, a fully inflected lexical form in Korean has often been called a WORD-PHRASE (`'ece1'`). To accurately describe this characteristic of Korean morphology, each word-phrase is not only assigned with a POS tag, but also annotated for morphological analysis. Our Treebank uses two major types of POS tags: 14 content tags and 15 function tags. For each word-phrase, the base form (stem) is given a content tag, and its inflections are each given a function tag. Word phrases are separated by a space, and within a word-phrase, the base form and inflections are separated by a plus sign (+). In addition to POS tags, the tagset also consists of 5 punctuation tags. An example of tagged sentence is given in (??).<sup>2</sup>

- (1) a. Raw text:  
 cacwu thongsinmangul wunyonghanta.  
 frequently com\_net-Acc operate-Decl  
 '(We) operate communications network frequently.'

---

<sup>2</sup>NNC and NNX are noun tags, PAD, PCA and PAU are noun inflectional tags, ADV is an adverb tag, XSV is a verbalizing suffix tag, EFN is a sentence final ending tag, and SFN is a punctuation tag. For a detailed description of the tagset, see han-tb1.

b. Tagged text:

cacwu/ADV thongsinmang/NNC+ul/PCA wun Yong/NNC+ha/XSV+nta/EFN  
./SFN

The main criterion for tagging and also for resolving ambiguity is syntactic distribution: i.e., a word may receive different tags depending on the syntactic context in which it occurs. For example, ‘akka’ (*some time ago*) is tagged as a common noun (NNC) if it modifies another noun, and is tagged as an adverb (ADV) if it modifies a verb.

- (2) a. akka/ADV ka/VV+ass/EPF+ta/EFN  
some\_time\_ago go-Past-Decl  
b. akka/NNC+uy/PCA yaksok/NNC  
some\_time\_ago-Gen promise

One important decision we had to make was whether to treat case postpositions and verbal endings as a bound morpheme or as a separate word. The decision we make on this issue would have consequences on syntactic bracketing as well. If we were to annotate them as separate words, it would be only natural to bracket them as independent syntactic units, which project their own functional syntactic nodes. Although some may favor this approach as theoretically more sound, from a descriptive point of view, they are more like bound morphemes, in that they are rarely separated from stems in written form, and native speakers of Korean share the intuition that they can never stand alone meaningfully in both written and spoken form. To reflect this intuition, we have chosen to annotate the inflections as bound morphemes assigning them each with a function tag.

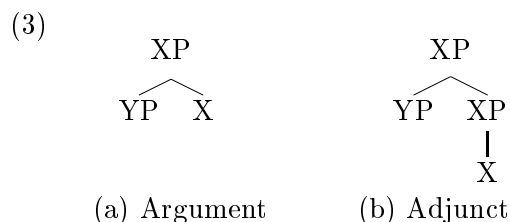
## 2.2 Syntactic bracketing

Penn Korean Treebank uses phrase structure annotation for syntactic bracketing. Similar phrase structure annotation schemes were also used by Penn English Treebank (mitch,penntreebank), Penn Middle English Treebank (krochtaylor95 and Penn Chinese Treebank xia2000-LREC). This annotation is preferable to a pure dependency annotation because it can encode richer structural information. For instance, some of the structural information that a phrase structure annotation can encode, while dependency annotation cannot, are (i) phrasal level node labels such as VP and NP; (ii) explicit representation of empty arguments; (iii) distinction between complementation and adjunction; and (iv) use of traces for displaced constituents.

Although having traces and empty arguments may be controversial, it has been shown in collins97,collins99 that such rich structural annotation is crucial in improving the efficiency of stochastic parsers that are trained on Treebanks. Moreover, it has been shown in rambowjoshi97 that a complete mapping from dependency structure to phrase structure cannot be done, although the other direction is possible. This means that a phrase structure Treebank can always be converted to a dependency Treebank if necessary, but not the other way around.

The bracketing tagset of our Treebank can be divided into four types: (i) POS tags for head-level annotation (e.g., NNC, VV, ADV); (ii) syntactic tags for phrase-level annotation (e.g., NP, VP, ADVP); (iii) function tags for grammatical function annotation (e.g., -SBJ for subject, -OBJ for object, -ADV for adjunct); and (iv) empty category tags for dropped arguments (\*pro\*), traces (\*T\*), and so on.

In addition to using function tags, arguments and adjuncts are distinguished structurally as well. If YP is an internal argument of X, then YP is in sister relation with Y, as represented in (??a). If YP is an adjunct of X, then YP adjoins onto XP, a projection of X, as in (??b).



The syntactic bracketing of example (??) is given in the left side of Table ???. This example contains an empty subject, which is annotated as (NP-SBJ \*pro\*). The object NP ‘thongsinmang/NNC+ul/PCA’ is assigned with -OBJ function tag, and since it is an argument of the verb, it is structurally a sister of the verb. The adverb ‘cacwu’ is an adjunct of the verb, and so it is adjoined to the VP, the phrasal projection of the verb.

An example sentence with a displaced constituent is given in (??). In this example, the object NP ‘kwenhanul’ appears before the subject, while its canonical position is after the subject. Displacement of argument NPs is called SCRAMBLING.

- (4) kwenhanul    nwuka    kaciko    issci?  
 authority-Acc who-Nom have    be  
 ‘Who has the authority?’

(S (NP-SBJ \*pro\*)  
 (VP (ADVP cacwu/ADV)  
 (VP (NP-OBJ thongsinmang/NNC+ul/PCA)  
 (VV wunyong/NNC+ha/XSV+nta/EFN))))  
 ./SFN)

(S (NP-OBJ-1 kwenhan/NNC+ul/PCA)  
 (S (NP-SBJ nwukwu/NPN+ka/PCA)  
 (VP (VP (NP-OBJ \*T\*-1)  
 kaci/VV+ko/EAU  
 iss/VX+ci/EFN))  
 ?/SFN )

Table 1: Syntactic bracketing examples

In our annotation in the right side of Table ??, the object is adjoined to the main clause (S), and leaves a trace (\*T\*) in its original position which is coindexed with it.

A potential cause for inconsistency is making argument/adjunct distinction. To ensure consistency in this task, we extracted all the verbs and adjectives from the corpus, and created what we call a PREDICATE-ARGUMENT LEXICON, based on Korean dictionaries, usages in the corpus and our own intuition. This lexicon lists verbs and adjectives with their subcategorization frame. For instance, the verb ‘wunyongha’ (*operate*) is listed as a transitive verb requiring a subject and object obligatory arguments. We also have a notation for optional arguments for some verbs. For instance, in (??), although we have clear intuition that ‘ecey’ (*yesterday*) is an adjunct and ‘wulinun’ (*we*) is the subject argument of ‘ka’ (*to go*), hakkyoey (‘to school’) seems to be a mixture of an argument and adjunct. This intuition is represented in the predicate-argument lexicon by listing an obligatory subject argument and a locative optional argument for ‘ka’ (*to go*).

- (5) wulinun ecey      hakkyoey kassta.  
 we-Top yesterday school-to go-Past-Decl  
 ‘We went to school yesterday.’

In syntactic bracketing, while an obligatory arguments are annotated with -SBJ or -OBJ function tag, if a sentence contains an optional argument, it is annotated with -COMP function tag. Moreover, a missing oblig-

atory argument is annotated as an empty argument, but a missing optional argument does not count as an empty argument.

Another potential cause for inconsistency is handling ambiguous sentences. To avoid such inconsistencies, we have classified the types of ambiguities, and specified the treatment of each type in the bracketing guidelines. For example, a subset of Korean adverbs can occur either before or after the subject. When the subject is phonologically empty, in principle, the empty subject can be marked either before or after the adverb without difference in meaning if there is no syntactic/contextual evidence for favoring one analysis over the other. In this case, to avoid any unnecessary inconsistencies, a ‘default’ position for the subject is specified and the empty subject is required to be put before the adverb. An example annotation is already given in Table ??.<sup>3</sup>

### 3 Annotation process

The annotation proceeded in three phases: the first phase was devoted to morphological analysis and POS tagging, the second phase to syntactic bracketing and the third phase to quality control.

#### 3.1 Phase I: morphological analysis and POS tagging

We used an off-the-shelf Korean morphological analyzer (jtyoon99) to facilitate the POS tagging and morphological analysis. We ran the entire corpus through this morphological analyzer and then automatically converted the output POS tags to the set of POS tags we had defined. We then hand-corrected the errors in two passes. The first pass took roughly two months to complete by two annotators. During this period, various morphological issues from the corpus were discussed in weekly meetings and guidelines for annotating them were decided and documented. In the second pass, in about a month, each annotator double-checked and corrected the files annotated by the other annotator.

#### 3.2 Phase II: Syntactic bracketing

The syntactic bracketing also went through two passes. The first pass took about 6 months to complete by three annotators, and the second pass took about 4 months to complete by two annotators. In the second pass, the annotators double-checked and corrected the bracketing done during the first

---

<sup>3</sup>See han-tb2 for the documentation of our syntactic bracketing guidelines.

pass. Phase II took much longer than Phase I because all the syntactic bracketing had to be done from scratch. Moreover, there were far more syntactic issues to be resolved than morphological issues. As in Phase I, weekly meetings were held to discuss and investigate the syntactic issues from the corpus and annotation guidelines were decided and documented accordingly. The bracketing was done using the already existing emacs-based interface developed for Penn English Treebanking (described in mitch), which we customized for Korean Treebanking. Using this interface helped to avoid bracketing mismatches and errors in syntactic tag labeling.

### 3.3 Phase III: Quality control

In order to ensure accuracy and consistency of the corpus, an entire third phase of the project was devoted to quality control. During this period, several full-scale examinations on the whole corpus were conducted, checking for inconsistent POS tag and illegal syntactic bracketings. LexTract was used to detect formatting errors (xia2000).

**Correcting POS tagging errors** Errors in POS tagging can be classified into three types: (a) assignment of an impossible tag to a morpheme (b) ungrammatical sequence of tags assigned to a word-phrase, and (c) wrong choice of a tag (sequence) candidate in the presence of multiple tag (sequence) candidates.

Type (a) was treated by compiling a tag dictionary for the entire list of morphemes occurring in the corpus. For closed lexical categories such as verbal endings, postposition markers and derivational suffixes, all of them were examined to ensure that they are assigned with correct tags. For open-set categories such as nouns, adverbs, verbs and so on, only those word-tag combinations exhibiting a low frequency count were individually checked.

Treating type (b) required knowledge in morphosyntax of Korean. First, a table of all tag sequences and their frequencies in the corpus was compiled, as shown in Table ??.

Those tag sequences found less than 3 times were all manually checked for their grammaticality, and corrected if found illegal. As a next step, a set of hand-crafted morphotactic rules were created in the form of regular expressions. Starting from the most rigorous patterns, we checked the tag sequences against the patterns already incorporated in the set of grammatical rules, expanding the set as needed or invalidating a tag sequence according to the outcome.

Rank	Count	Count%	Total%	Entry
1	8647	15.85	15.85	NNC
2	5606	10.28	26.14	NNC+PCA
3	5083	9.32	35.46	SFN
...	...	...	...	...
221	1	0.00	99.99	NNC+XSF+CO+EPF+ENM
221	1	0.00	100	NNC+XSV+EPF+EFN+PCA

Table 2: Frequency of tag sequences

Type (c), assignment of a wrong tag in the case of ambiguity, cannot be handled by looking at the morphemes by themselves, but syntactic context must be considered: therefore this type of problems were treated along with other illegal syntactic structures.

**Correcting illegal syntactic structures** To correct errors in syntactic bracketing, we targeted each local tree structure (parent node + daughter nodes). To do this, all local tree structures were extracted in the form of context-free rules (Table ??). For local trees with a lexical daughter node, the lexical information was ignored and only POS information on the node was listed in the rule.

Rank	Count	Count%	Total%	Entry
1	5993	7.72	7.72	S → NP-SBJ VP
2	4079	5.26	12.98	NP-SBJ → *pro*
3	2425	3.12	16.11	ADVP → ADV
...	...	...	...	...
1394	1	0.00	99.99	ADJP → VJ+EPF+EFN+PAU
1394	1	0.00	100	ADJP → S NP-ADV ADVP ADJP

Table 3: Frequency of context-free rules

The next step taken was to define the set of context-free rules for Korean in the form of regular expression. For each possible intermediate node label (phrasal categories as S, NP, VP and a few lexical categories such as VV and VJ) on the lefthand side of the rule, its possible descendant node configuration was defined as a regular expression, as seen in (??):

- (6) a. VP (shown in part):  
 (NP-OBJ(-LV)? | NP-COMP(-LV)?  
 | S-COMP | S-OBJ )+ VV\S\*
- b. VV:  
 NNC(\+XSF)?\+XSV  
 |\~VV\S\* VV\S\*\$ | (VV )\*(ADCP )?VV

Example (??a) shows that a local tree with VP as the parent node can have as its daughter nodes any numbers of NP-OBJ, NP-COMP, S-COMP or S-OBJ nodes followed by a VV node, which is the head.

As with the case of word-internal tag sequences, the most frequent context-free rules were examined and incorporated into the set of rules first, and this set gradually grew as more and more rules were examined and decided to be included in the rule set or rejected to be corrected later. As a result, a large number of illegal syntactic bracketings were identified and corrected. Particularly frequent types of syntactic tagging errors were: (a) redundant phrasal projections (i.e.  $VP \rightarrow VP$ ), (b) missing phrasal projections, and (c) misplaced or ill-scoped modifying elements such as relative clauses and adverbial phrases/clauses.

**Corpus search** We compiled a list of error-prone or difficult syntactic constructions that had been observed to be troublesome and confusing to annotators, and used corpus search tools (randall2000) to extract sentence structures containing each of them from the Treebank. Each set of extracted structures were then examined and corrected. The list of constructions we looked at in detail include relative clauses, complex noun phrases, light verb constructions, complex verbs, and coordinate structures. By doing a construction by construction check of the annotation, not only were we able to correct errors but also enhance the consistency of our annotation.

## 4 Statistics on the size of corpus

In this section, we present some quantitative aspects of the Penn Korean Treebank corpus. The corpus is a relatively small one with 54,528 words and 5,083 sentences, averaging 9.158 words per sentence. A total of 10,068 word types are found in the corpus, therefore the measured type/token ratio (TTR) is 0.185. This figure is high compared to English texts (average English written document is said to have a TTR of 0.15 \*\*\*find good reference\*\*\*). However, for languages with rich agglutinative morphology such

as Korean, even higher type/token ratios are not uncommon. For comparison, a comparably sized portion (54,547 words) of the ETRI corpus, an annotated corpus with POS tags, was taken and analyzed.<sup>4</sup> This set contained 19,889 word types, almost double the size of that of the Penn Korean Treebank, as shown in Table ??.

	word		
	token	type	type/token ratio
Trebank	54,528	10,068	0.185
ETRI	54,547	19,889	0.364
	morpheme		
	token	type	type/token ratio
Trebank	93,148	3,555	0.038
ETRI	101,100	8,734	0.086

Table 4: Type/token ratios of two corpora

Taking individual morphemes, rather than words in their fully inflected forms, as the evaluation unit, the ratio becomes much smaller: Penn Korean Treebank yields the morpheme type/token ratio of 0.038 (93,148 tokens and 3,555 types). Compared to the portion of ETRI corpus, we can see that Penn Korean Treebank still shows a lower ratio: ETRI corpus showed the morpheme type/token ratio of 0.086 (101,100 morpheme tokens and 8,734 unique morpheme types).

The result suggests that Penn Korean Treebank, aimed to be a domain-specific corpus in the military domain, is highly homogeneous and low in complexity at least in terms of its lexical content. ETRI corpus, on the other hand, consists of texts from different genres including novels, news articles and academic writings, hence the higher counts of lexical entries per word token. In our future work, we hope to expand the Treebank corpus in order to achieve a broader and more general coverage.

<sup>4</sup>Total of 12 files: essay01.txt, expl10.txt, expl34.txt, news02.txt, newsp05.txt, newsp12.txt, newsp15.txt, newsp16.txt, novel03.txt, novel13.txt, novel15.txt and novel19.txt. For fair comparison, the POS annotated text was re-tokenized to suit the Penn Korean Treebank standards.

## 5 Evaluation

For evaluating the consistency and accuracy of the Treebank, we used Evalb software that produces three metrics, bracketing precision, bracketing recall and numbers of crossing brackets, as well as tagging accuracy.

For the purposes of evaluation, we randomly selected 10% of the sentences from the corpus in the beginning of the project and saved them to a file. These sentences were then POS tagged and bracketed just like any other sentences in the corpus. After the first pass of syntactic bracketing, however, they were double annotated by two different annotators. We also constructed a Gold Standard annotation for these test sentences. We then ran Evalb on the two annotated files produced by the two different annotators to measure the inter-annotator consistency. Evalb was also run on the Gold Standard and the annotation file of the 1st annotator, and on the Gold Standard and the annotation file of the 2nd annotator to measure the individual annotator accuracy. Table ?? shows the accuracy of each annotator compared to the Gold Standard under *1st/gold* and *2nd/gold* column headings, and the inter-annotator consistency under *1st/2nd* column heading. It shows that all the measures are well over 95%, tagging accuracy reaching almost 100%. These measures indicate that the quality of the Treebank is more than satisfactory.

	Consistency	Accuracy	
	1st/2nd	1st/gold	2nd/gold
Recall	96.60	97.69	98.84
Precision	97.97	98.89	98.84
No Crossing	95.89	97.57	97.53
Tagging	99.72	99.99	99.77

Table 5: Inter-annotator consistency and accuracy of the Treebank

Most of the inter-annotator inconsistencies belonged to one of the following types:

- In coordinated sentences with empty subject and empty object, whether the level of coordination is VV, VP or S;
- Whether a sentence has empty object argument or not;
- Whether a noun modified by a clause is a relative clause construction or a complex NP;

- Whether a verb is a light verb or a regular verb;
- In a complex sentence in which the subject of the matrix clause and the subordinate clause are coreferential, whether a topic marked NP is the subject of the matrix clause or the subordinate clause;
- In a sentence with a topic marked object NP and an empty subject, whether the object NP has undergone scrambling over the empty subject or not;
- For an NP with an adverbial postposition<sup>5</sup>, whether it is an argument or an adjunct;
- When an adverb precedes another adverb which in turn precedes a verb, whether the first adverb modifies the adverb or the verb.

After the evaluation was done, as a final cleanup of the Treebank, using corpus search tools, we extracted and corrected structures that belong to those that may potentially lead to the types of inconsistencies described above.

## 6 Conclusion

We have described in detail the annotation process as well as the methods we used to ensure inter-annotator consistency and annotation accuracy in creating a 54K word Korean Treebank.<sup>6</sup> We have also discussed the major principles employed in developing POS tagging and syntactic bracketing guidelines. Despite the small size of the Treebank, we were able to successfully train a morphological tagger (95.78%/95.39% precision/recall) and a parser (73.45% dependency accuracy) using the data from the Treebank. They were incorporated to Korean/English machine translation system which were jointly developed by University of Pennsylvania and CoGenTex (han-amta00).

We plan to release the Treebank in the near future making it available to the wider community. The corpus we used for the Korean Treebank is originally from a Korean/English parallel corpora, and we are currently in the process of creating a Korean/English parallel Treebank by treebanking

---

<sup>5</sup>Adverbial postpositions correspond to English prepositions in function, e.g., `-eykey` (*to*), `-lopyte` (*from*), `-ey` (*in*), etc.

<sup>6</sup>Information on our Penn Korean Treebank can be found in [www.cis.upenn.edu/~xtag/koreantag/](http://www.cis.upenn.edu/~xtag/koreantag/), including POS tagging and syntactic bracketing guidelines as well as a sample bracketed file.

the English side and aligning the two Treebanks. We would also like to expand the size and coverage of the corpus by treebanking newswire corpora, employing as rigorous an annotation methodology as we did for the 54K Treebank. We hope to speed up the annotation process by automaticizing the annotation process as much as possible (Cf., along the lines described in negra for NEGRA corpus at the University of Saarbrücken), incorporating a parser as well as a tagger to the annotation interface.

## Acknowledgements

We thank Aravind Joshi, Tony Kroch and Fei Xia for valuable discussions on many occasions. Special thanks are due to Owen Rambow, Nari Kim and Juntae Yoon for discussions in the initial stage of the project. The work reported in this paper was supported by contract DAAD 17-99-C-0008 awarded by the Army Research Lab to CoGenTex, Inc., with the University of Pennsylvania as a subcontractor, NSF Grant -VerbNet, IIS 98-00658, and DARPA Tides Grant N66001-00-1-8915.