Proceedings of the 3rd International Conference on Language Resources and Evaluation 2002 (LREC-2002).

Development and Evaluation of a Korean Treebank and its Application to NLP

Chung-hye Han*, Na-Rare Han†, Eon-Suk Ko†, Martha Palmer‡

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

†Dept. of Linguistics
University of Pennsylvania
619 Williams Hall
Philadelphia, PA 19104, USA
{nrh,esko}@ling.upenn.edu

‡Dept. of Computer Information and Science
University of Pennsylvania
256 Moore School
Philadephia, PA 19104, USA
mpalmer@linc.cis.upenn.edu

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 are presented. The evaluation on the quality of the Treebank and some of the NLP applications under development using the Treebank are also presented.

1. Introduction

With growing interest in Korean language processing, numerous natural language processing (NLP) tools for Korean, such as part-of-speech (POS) taggers, morphological analyzers and parsers, have been developed. This progress was possible through the availability of large-scale raw text corpora and POS tagged corpora (ETRI, 1999; Yoon and Choi, 1999a; Yoon and Choi, 1999b). However, no large-scale bracketed corpora are currently available to the public, although efforts have been made to develop guidelines for syntactic annotation (Lee et al., 1996; Lee et al., 1997). As a step towards addressing this issue, we built a 54-thousand-word[1] Korean Treebank using a phrase structure annotation at the University of Pennsylvania, creating the Penn Korean Treebank. At the same time, we also developed 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 a 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 §3., including various methods we used to detect and correct annotation errors. §4. presents some statistics on the size of the corpus. §5. discusses the results of the evaluation on the Treebank, and §6. presents some of the NLP applications we did so far using the Treebank.

2. Guideline development

The guiding principles employed in developing the annotation guidelines were theory-neutralness (whenever possible), descriptive accuracy and consistency. 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 copora by such institutions as KAIST and ETRI (ETRI, 1999; Lee et al., 1996; Lee et al., 1997; Yoon and Choi, 1999a; Yoon and Choi, 1999b). Ideally, complete guidelines should be available to the annotators before annotation begins. However, linguistic problems posed by corpora are 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

[1]This word count is computed on tokenized texts and includes symbols.

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 the 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 ('어절'). 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 a tagged sentence is given in (1).[2]

(1) a. Raw text:

자주 통신망을 운용한다.
frequently com_net-Acc operate-Decl

'(We) operate communications network frequently.'

b. Tagged text:
자주/ADV 통신망/NNC+을/PCA
운용/NNC+하/XSV+ㄴ다/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, '아까' (*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. 아까/ADV 가/VV+았/EPF+다/EFN
some_time_ago go-Past-Decl

b. 아까/NNC+의/PCA 약속/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 a function tag.

[2]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 and Han, 2001).

2.2. Syntactic bracketing

The Penn Korean Treebank uses phrase structure annotation for syntactic bracketing. Similar phrase structure annotation schemes were also used by the Penn English Treebank (Marcus et al., 1993; Bies et al., 1995), the Penn Middle English Treebank (Kroch and Taylor, 1995) and the Penn Chinese Treebank, (Xia et al., 2000b). 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 readily encodes, which dependency annotations typically do not, 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 (Collins, 1997; Collins et al., 1999) 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 (Rambow and Joshi, 1997) 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 X, as represented in (3a). If YP is an adjunct of X, then YP adjoins onto XP, a projection of X, as in (3b).

(3)

(a) Argument (b) Adjunct

The syntactic bracketing of example (1) is given in the first tree of Figure 1. This example contains an empty subject, which is annotated as (NP-SBJ *pro*). The object NP '통신망/NNC+을/PCA' is assigned the -OBJ function tag, and since it is an argument of the verb, it is structurally a sister of the verb. The adverb '자주' is an adjunct of the verb, and so it is adjoined to the VP, the phrasal projection of the verb.

```
(S (NP-SBJ *pro*)
   (VP (ADVP 자주/ADV)
       (VP (NP-OBJ 통신망/NNC+을/PCA)
           (VV 운용/NNC+하/XSV+ㄴ다/EFN))))
   ./SFN)


(S (NP-OBJ-1 권한/NNC+을/PCA)
   (S (NP-SBJ 누구/NPN+가/PCA)
      (VP (VP (NP-OBJ *T*-1)
              가지/VV+고/EAU)
          있/VX+지/EFN))
   ?/SFN)
```

Figure 1: Examples of syntactic bracketing

An example sentence with a displaced constituent is given in (4). In this example, the object NP '권한을' appears before the subject, while its canonical position is after the subject. Displacement of argument NPs is called SCRAMBLING.

(4) 권한을 누가 가지고 있지?
authority-Acc who-Nom have be
'Who has the authority?'

In our annotation in the second tree of Figure 1, 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 frames. For instance, the verb '운용하' (*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 (5), it is not clear whether '학교에' (*to school*) is an argument or an adjunct, whereas '어제' (*yesterday*) and '우리는' (*we*) seem to offer clear intuition as to their adjunct and argument status, respectively. This is resolved by listing such categories as a locative optional argument for '가' (*to go*) in the predicate-argument lexicon.

(5) 우리는 어제 학교에 갔다.
we-Top yesterday school-to go-Past-Decl
'We went to school yesterday.'

In syntactic bracketing, while obligatory arguments are annotated with -SBJ or -OBJ function tag, if a sentence contains an optional argument, it is annotated with a -COMP function tag. Moreover, a missing obligatory 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 syntactically 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 Figure 1.[3]

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 (Yoon et al., 1999) 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, which was undertaken in about a month from the completion of the first phase, 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 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 (Marcus et al., 1993)), 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, the 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

[3] See (Han et al., 2001) for the documentation of our syntactic bracketing guidelines.

for inconsistent POS tags and illegal syntactic bracketings. LexTract was used to detect formatting errors (Xia et al., 2000a).

3.3.1. 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 of Korean morphosyntax. First, a table of all tag sequences and their frequencies in the corpus was compiled, as shown in Table 1.

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 morphotactic rules, expanding the set as needed or invalidating a tag sequence according to the outcome.

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

3.3.2. 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 2). 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.

The next step taken was to define the set of context-free rules for Korean. 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):

(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 (6a) 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.

3.3.3. 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 (Randall, 2000) 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, we were able to not only correct errors but also enhance the consistency of our annotation.

4. Statistics on the size of the 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 rather high at 0.185. 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 selected and analyzed.[4] This set contained 19,889 word types, almost double the size of that of the Penn Korean Treebank, as shown in Table 3.

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

Table 3: 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: the Penn Korean Treebank yields a

[4]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.

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 1: Frequency of tag sequences

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 2: Frequency of context-free rules

morpheme type/token ratio of 0.038 (93,148 tokens and 3,555 types). Compared to the same portion of the ETRI corpus, we can see that the Penn Korean Treebank still shows a lower ratio: the ETRI corpus showed a morpheme type/token ratio of 0.086 (101,100 morpheme tokens and 8,734 unique morpheme types).

These results suggest that the Penn Korean Treebank, as a domain-specific corpus in the military domain, is highly homogeneous and low in complexity at least in terms of its lexical content. The 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.

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 4 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 4: 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 an empty subject and an empty object, whether the level of coordination is VV, VP or S;
- Whether a sentence has an 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[5], 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. Applications of the Treebank

6.1. Morphological tagger

We trained a morphological tagger on 91% of the 54K Korean Treebank and tested it on 9% of the Treebank (Han, 2002). The tagger/analyzer takes raw text as input and returns a lemmatized disambiguated output in which for each word, the lemma is labeled with a POS tag and the inflections are labeled with inflectional tags. This system is based on a simple statistical model combined with a corpus-driven rule-based approach, comprising a trigram-based tagging component and a morphological rule application component.

The tagset consists of possible tag sequences (e.g., NNC+PCA, VV+EPF+EFN) extracted from the Treebank. Given an input sentence, each word is first tagged with a tag sequence. Tags for unknown words are then updated using inflectional templates extracted from the Treebank. A few example templates are listed in Table 5.

었습니다	VV+EPF+EFN
었습니다만	VV+EPF+ECS
었었기	VV+EPF+ENM
었으니까	VV+EPF+ECS
었으며	VV+EPF+ECS
었으므로	VV+EPF+ECS

Table 5: Example of Inflectional Templates

Using an inflection dictionary and a stem dictionary extracted from the Treebank, the lemma and the inflections are then identified, splitting the inflected form of the word into its constituent stem and affixes. This approach yielded 95.01%/95.30% recall/precision on the test data. An example input and output are shown below. The morphological tagger assigns POS tags and also splits the inflected form of the word into its constituent stem and inflections.

(7) a. Input:
제가 관측 사항을 보고했습니다.

b. Output:
제/NPN+가/PCA
관측/NNC
사항/NNC+을/PCA
보고하/VV+었/EPF+습니다/EFN
./SFN

[5]Adverbial postpositions correspond to English prepositions in function, e.g., '-에게' (*to*), '-로부터' (*from*), '-에' (*in*), etc.

6.2. Parser

The Treebank has been used to train a statistical parser using a probabilistic Tree Adjoining Grammar (TAG) model (Sarkar, 2002). The parser uses, as training data, TAG derivations automatically extracted from the Treebank with Xia's (Xia et al., 2000a) LexTract.

In a probabilistic TAG (Schabes, 1992; Resnik, 1992), each word in the input sentence is assigned a set of trees, called elementary trees that it has selected in the training data. Each elementary tree has some word (called the ANCHOR) in the input sentence as a node on the frontier. A derivation proceeds as follows: one elementary tree is picked to be the start of the derivation. Elementary trees are then added to this derivation using the operations of substitution and adjunction. Each tree added in this step can be recursively modified via subsequent operations of substitution and adjunction. Once all the words in the input sentence have been recognized, the derivation is complete. The parser outputs a derivation tree, a record of how elementary trees are combined to generate a sentence, and also a derived tree (read off from the derivation tree) which corresonds to the bracketed structure of a sentence.

The parser is interfaced to the morphological tagger described in §6.1. to avoid the sparse data problems likely to be caused by the highly agglutinative nature of words in Korean. The parser is able to use information from component parts of the words that the morphological tagger provides. With this method, we achieved 75.7% accuracy of TAG derivation dependencies on the test set from the Treebank. An example parser output of a derivation is given in Figure 2. The index numbers in the first column, and the last column of the table represent the dependencies between words. For instance, '모든 (*motun*)' has index 0 and it is dependent on the word indexed with 2 '대호+는 (*tayho+nun*)'. '바뀌+게 (*pakwi+key*)' is the root of the derivation, marked by TOP. The morpheme boundaries in the words in the 2nd column are marked with + sign. The 3rd column contains the tag sequence of the word, and the 4th column lists the names of the elementary tree anchored by the word.

7. 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.[6] 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 into a Korean/English machine translation system which was jointly developed by the University of Pennsylvania and CoGenTex (Han et al., 2000; Palmer et al., 2002).

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

Index	Word	POS tag (morph)	Elem Tree	Anchor Label	Node Address	Subst/Adjoin into (Index)
0	모든	DAN	βNP*=1	anchor	root	2
1	호철	NNC	βNP*=1	anchor	root	2
2	대호+는	NNC+PAU	αNP=0	anchor	0	6
3	매일	ADV	βVP*=25	anchor	1	6
4	24	NNU	βNP*=1	anchor	0	5
5	시+에	NNX+PAD	βVP*=17	anchor	1	6
6	바뀌+게	VV+ECS	αS-NPs=23	anchor	-	TOP
7	되+지요	VX+EFN	βVP*=13	anchor	1	6
8	.	SFN	-	-	-	-

Figure 2: Example of derivation of a sentence reported by the statistical parser

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 copora, and we have recently finished creating a Korean/English parallel Treebank by treebanking the English side and aligning the two Treebanks. We would 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 automating the annotation process as much as possible (Cf., along the lines described in (Skut et al., 1997) for NEGRA corpus at the University of Saarbrüken), 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. We would also like to thank Myuncheol Kim, Chulwoo Park and Heejong Yi for their participations in various parts 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.

8. References

Ann Bies, Mark Ferguson, Karen Katz, and Robert MacIntyre. 1995. Bracketing guidelines for treebank ii style penn treebank project.

Michael Collins, Jan Hajičč, Lance Ramshaw, and Christoph Tillmann. 1999. A statistical parser for czech. In *Proceedings for the 37th Annual Meeting of the Association for Comoputational Linguistics*, pages 505–512. Association for Computational Linguistics.

Michael Collins. 1997. Three generative, lexicalised models for statistical parsing. In *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics*, Madrid, Spain, July.

ETRI. 1999. 품사 태그 지침서. Technical report, 지식정보연구부, 컴퓨터-소프트웨어 기술 연구소, 한국전자통신연구원, Taejun, Korea. ETRI POS tagset guidelines.

Chung-hye Han and Na-Rae Han. 2001. Part of speech tagging guidelines for Penn Korean Treebank. Technical Report IRCS Report 01-09, IRCS, University of Pennsylvania. ftp://ftp.cis.upenn.edu/pub/ircs/tr/01-09/.

Chung-hye Han, Benoit Lavoie, Martha Palmer, Owen Rambow, Richard Kittredge, Tanya Korelsky, Nari Kim, and Myunghee Kim. 2000. Handling structural divergences and recovering dropped arguments in a Korean/English machine translation system. In John S. White, editor, *Envisioning Machine Translation in the Information Future*, pages 40–53. Springer-Verlag. Proceedings of AMTA 2000.

Chung-hye Han, Na-Rae Han, and Eon-Suk Ko. 2001. Bracketing guidelines for Penn Korean Treebank. Technical Report IRCS Report 01-10, IRCS, University of Pennsylvania. ftp://ftp.cis.upenn.edu/pub/ircs/tr/01-10/.

Chung-hye Han. 2002. A morphological analyzer/tagger for Korean: statistical tagging combined with corpus-based morphological rule application. Ms. Simon Fraser University.

Anthony Kroch and Ann Taylor. 1995. The Penn-Helsinki parsed corpus of Middle English.

Kongjoo Lee, Jaehoon Kim, Pyengkyu Cang, Kisun Choi, and Kilchang Kim. 1996. Syntactic tag set for Korean syntactic tree tagged corpus. Technical Report CS-TR-96-102, Department of Computer Science, KAIST. Written in Korean.

Kongjoo Lee, Pyengkyu Cang, and Kilchang Kim. 1997. Bracketing guidelines for Korean syntactic tree tagged corpus version i. Technical Report CS-TR-97-112, Department of Computer Science, KAIST. Written in Korean.

Mitch Marcus, Beatrice Santorini, and M. Marcinkiewicz. 1993. Building a large annotated corpus of English. *Computational Linguistics*, 19(2):313–330.

Martha Palmer, Chung-hye Han, Anoop Sarkar, and Ann Bies. 2002. Integrating Korean analysis components in a modular Korean/English machine translation system. Ms. University of Pennsylvania and Simon Fraser University.

Owen Rambow and Aravind K. Joshi. 1997. A formal look at dependency grammars and phrase structure grammars with special considertaion of word-order phenomena. In L. Wenner, editor, *Recent Trends in Meaning-Text The-*

ory. John Benjamin, Amsterdam, Philadelphia.

Beth Randall, 2000. *CorpusSearch User's Manual*. University of Pennsylvania. http://www.cis.upenn.edu/ brandall/CSManual/.

Philip Resnik. 1992. Probabilistic tree-adjoining grammars as a framework for statistical natural language processing. In *Proceedings of COLING '92*, volume 2, pages 418–424, Nantes, France.

Anoop Sarkar. 2002. *Statistical Parsing Algorithms for Lexicalized Tree Adjoining Grammars*. Ph.D. thesis, University of Pennsylvania.

Yves Schabes. 1992. Stochastic lexicalized tree-adjoining grammars. In *Proceedings of COLING '92*, volume 2, pages 426–432, Nantes, France.

Wojciech Skut, Brigitte Krenn, Thorsten Brants, and Han Uszkoreit. 1997. An annotation scheme for free word order languages. In *Proceedings of the Fifth Conference on Applied Natural Language Processing (ANLP)*, Washington, D.C.

Fei Xia, Martha Palmer, , and Aravind Joshi. 2000a. A uniform method of grammar extraction and its applications. In *Proceedings of the EMNLP 2000*.

Fei Xia, Martha Palmer, Nianwen Xue, Mary Ellen Okurowski, John Kovarik, Shizhe Huang, Tony Kroch, and Mitch Marcus. 2000b. Developing Guidelines and Ensuring Consistency for Chinese Text Annotation. In *Proceedings of the 2nd International Conference on Language Resources and Evaluation (LREC-2000)*, Athens, Greece.

Juntae Yoon and Kisun Choi. 1999a. Study on kaist corpus. Technical Report CS-TR-99-139, Department of Computer Science, KAIST. Written in Korean.

Juntae Yoon and Kisun Choi. 1999b. Study on pos tagged corpus for Korean. Technical Report CS-TR-99-138, Department of Computer Science, KAIST. Written in Korean.

Juntae Yoon, C. Lee, S. Kim, and M. Song. 1999. Morphological analyzer of Yonsei univ., Morany: Morphological analysis based on large lexical database extracted from corpus. In *Proceedings of Korean Language Information Processing*. Written in Korean.