Natural Language Engineering (2022), **28**, pp. 199–222
doi:10.1017/S1351324920000479

CAMBRIDGE UNIVERSITY PRESS

ARTICLE

A note on constituent parsing for Korean

Mija Kim[1] and Jungyeul Park[2,*,†]

[1]Kyung Hee University, Seoul, South Korea and [2]University of Washington, Seattle, WA, USA
*Corresponding author. E-mail: jungyeul@uw.edu

(Received 15 August 2019; revised 12 October 2020; accepted 12 October 2020; first published online 10 November 2020)

Abstract

This study deals with widespread issues on constituent parsing for Korean including the quantitative and qualitative error analyses on parsing results. The previous treebank grammars have been accepted as being interpretable in the various annotation schemes, whereas the recent parsers turn out to be much harder for humans to interpret. This paper, therefore, intends to find the concrete typology of parsing errors, to describe how these parsers deal with sentences and to show their statistical distribution, using state-of-the-art statistical and neural parsers. For doing this work, we train and evaluate the phrase structure Sejong treebank using statistical and neural parsing systems and obtain results up to a 89.18% F_1 score, which outperforms previous constituent parsing results for Korean. We also define best practices for correct comparison to future work by proposing the standard corpus division for the Sejong treebank.

Keywords: Constituent parsing; Sejong treebank; Errors analysis; Korean

1. Introduction

Statistical and neural parsing techniques are widely known to be surprisingly powerful and compelling, in the sense that they are easily trained and evaluated if the syntactically annotated treebank is available. Their results, however, can be difficult to understand because parsing models by the state-of-the-art statistical and neural parsers are automatically induced from the treebank. Treebanks, collection of parsed and annotated corpora, constitute an essential resource for natural language processing in any given language (Abeillé 2003). The automatic syntactic analysis of sentences directly benefits syntactically annotated corpora. Currently, most of the state-of-the-art parsers use annotated syntactic information such as constituent structure or dependency relations to train probabilistic models. Several annotated phrase structure Korean treebanks have been created over the last two decades, such as the KAIST treebank (Choi *et al*. 1994), the Penn Korean treebank (Han *et al*. 2002) and the Sejong treebank (released in 2011).[a] Although there has been a shared task of constituent parsing for Korean using the KAIST treebank as part of the shared task on Statistical Parsing of Morphologically Rich Languages (Seddah *et al*. 2013; Seddah, Kübler, and Tsarfaty 2014), studies on constituent parsing for Korean using these treebanks have hardly been performed. For instance, there are just a few previous works that used these treebanks such as Chung *et al*. (2010), in which they are using the Penn treebank, and Choi *et al*. (2012) and Park *et al*. (2016) where they are using the Sejong treebank.

[†]Mija Kim and Jungyeul Park contributed equally.
[a]The Sejong treebank is available from the National Institute of Korean Language, https://www.korean.go.kr.

© The Author(s), 2020. Published by Cambridge University Press.

https://doi.org/10.1017/S1351324920000479 Published online by Cambridge University Press

1.1 Various phrase structures in Korean treebanks

Morphological analysis for Korean has been based on an *eojeol*, which has been considered as a basic unit of segmentation in Korean, delimited by blank spaces in the sentence. Given that Korean is an agglutinative language, joining content and functional morphemes of eojeols is very productive. The number of types for possible combined eojeols can be increased exponentially. For example, *unggaro* ('Ungaro') is a content morpheme (a proper noun) and a postposition *-ga* (nominative) is a functional morpheme. They form together a single eojeol *unggaro-ga* ('Ungaro + NOM'). For convenience sake, we add a hyphen (-) at the beginning of functional morphemes, such as *-ga* for NOM, to distinguish between content and functional morphemes. The nominative case markers *-ga* or *-i* may vary depending on the previous letter – vowel or consonant. A predicate *naseo-eoss-da* also consists of the content morpheme *naseo* ('become') and its functional morphemes, *-eoss* ('PAST') and *-da* ('IND'), respectively.

(1) 프랑스의 세계적인 의상 디자이너 엠마누엘 웅가로가
peurangseu-ui segye-jeok-i-n uisang dijaineo emmanuel unggaro-ga
France-GEN world class-REL fashion designer Emanuel Ungaro-NOM
실내 장식용 직물 디자이너로 나섰다.
silnae jangsik-yong jikmul dijaineo-ro naseo-eoss-da-.
interior decoration textile designer-AJT become-PAST-IND-.
'The world class French fashion designer Emanuel Ungaro became an interior textile designer'.

Meanwhile, different granularity levels for word segmentation have been independently proposed as different basic segmentation units for Korean treebanks based on the combination of content and functional morphemes. For example, the Sejong treebank (Figure 1(a)), the most widely used corpus for Korean, uses the eojeol as itself for the basic unit of analysis. That is, no tokenisation scheme in the treebank is proposed. The token in the Sejong treebank, therefore, may contain punctuation marks and other symbols along with words. The Sejong treebank explicitly utilises the binary structure using Chomsky normal form.

There are, however, treebanks which use an English-like tokenisation and multifurcating trees. Words in these corpora are already preprocessed, for example, the Penn Korean treebank (Han *et al*. 2002), in which punctuation marks are separated from words (Figure 1(b)). Moreover, the Penn Korean treebank particularly annotates these null elements in the sentence. Null elements in the Penn Korean treebank represent empty arguments (*pro*), traces (*T*), empty operators (*op*) and predicate deletions (*?*) (Han, Han, and Ko 2001). We also note that there are ver. 1.0 and 2.0 in the Penn Korean treebank, in which their characteristics of the corpus are completely different. The text of the former is from military manuals and the latter is from the Korean Newswire corpus which is a collection of Korean Press Agency news articles.

Government and Binding (GB) theory (Chomsky 1981, 1982, 1986) for Korean often proposed a syntactic analysis, in which the entire sentence depends on verbal endings. The KAIST treebank (Choi *et al*. 1994), the earliest Korean treebank introduced this type of analysis (Figure 1(c)). In the KAIST treebank, functional morphemes are explicitly separated from their content morphemes. The SPMRL 2013 and 2014 shared tasks used the *modified* version of the KAIST treebank for Korean parsing (Seddah *et al*. 2013, 2014). Whereas the original KAIST treebank explicitly separates postpositions and verbal endings from the phrase as in Figure 1(c), the modified KAIST treebank uses an eojeol-based tokenisation scheme as in the Sejong treebank.

Figure 1 shows different Korean treebanks using the different morphological segmentation granularity of the word in their phrase structures. Whereas there are studies on merging heterogeneous treebanks to improve parsing results such as Stymne *et al*. (2018) for dependency parsing, these three treebanks for Korean would still be difficult to be interchangeable because of the differences of their lexical and syntactic annotations. It is thus not easy to combine them together in order to obtain the bigger treebank.

https://doi.org/10.1017/S1351324920000479 Published online by Cambridge University Press

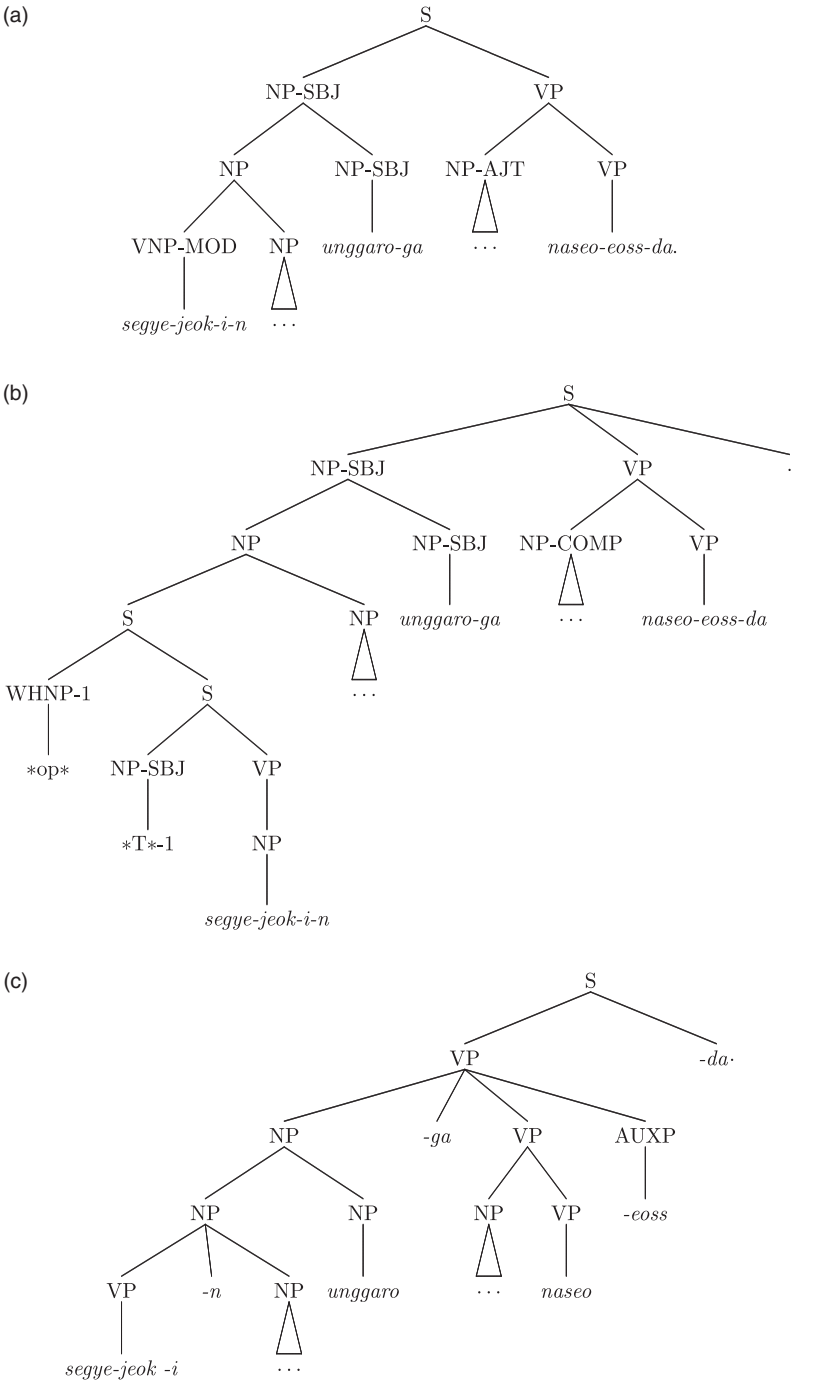


Figure 1. Different Korean treebanks using the different segmentation granularity levels. (a) The Sejong treebank structure (b) The Penn Korean treebank-like structure (c) The KAIST treebank-like structure.

https://doi.org/10.1017/S1351324920000479 Published online by Cambridge University Press

1.2 Previous work on constituent parsing for Korean

One of the earliest works on statistical constituent parsing for Korean is Hermjakob (2000) where he used 1187 treebank sentences taken from the Korean newspaper *Chosun Ilbo* ('Korea Daily News'). He implemented a shift-reduce parser based on his earlier work (Hermjakob and Mooney 1997) with syntactic and semantic features supported by background knowledge such as a time expression or a locative marker which are generally not covered by part-of-speech information. The average sentence length in the treebank was 21.0 words. He presented evaluation results with training sentences ranging from 32 to 1024, and the remaining 163 sentences of the treebank for testing. He got a relatively high-performance result with 91.0% (precision) and 89.8% (recall). Currently, the *Chosun* treebank is not available.

Sarkar and Han (2002) trained lexicalised tree adjoining grammars (LTAGs) from 4,653 sentences (91% of the corpus) of the early version of the Penn Korean treebank (ver. 1.0). LTAG grammars are extracted using LexTract (Xia *et al.* 2000). The statistical parser was implemented based on Sarkar (2002) and used three probabilistic models: one probability for suppertagging of the elementary tree with a lexical anchor (Bangalore and Joshi 1999) and two probabilities for substitution and adjunction operations where elementary trees can be combined with. The parser was then tested on 425 sentences (9%), and they obtained a 75.7% F_1 score.

Chung *et al.* (2010) used all 5010 sentences in the Penn Korean treebank (ver. 2.0) using 10-fold cross-validation for experiments. They obtained results up to a 89.56% F_1 score with the parsing techniques described in Post and Gildea (2009). They used an implementation of a CKY PCFG parser by Johnson (1998). Their treebank grammars are extracted by the Berkeley parser (Petrov and Klein 2007) using various factors including non-terminal granularity and null elements.

After Park (2006) introduced treebank grammars extracted from the early version of the Sejong treebank and its usability in the LTAG parser, Choi *et al.* (2012) used the Sejong treebank for constituent parsing. They proposed an entity-based preprocessing where certain set of morphemes are merged as a single entity, instead of a morpheme-based parsing as depicted in Figure 2(c), and improved parsing accuracy. The Sejong treebank does not distinguish special symbols from punctuation marks in the eojeol, both of which are annotated as a punctuation mark. During the entity-based preprocessing, entities, such as numbers with the decimal point (*3.14*), email addresses (*name@email.com*), web address (*http://www.web.info*), dates (*25/9/2017*), can be presented as a single token, while punctuation marks are still separated from the eojeol. Choi *et al.* (2012) introduced the Stanford parser that uses a generalised chart-based CKY algorithm (Klein and Manning 2003), the Bikel parser (Bikel 2004a) that introduces lexicalised rules based on the Collins parser (Collins 1999) and the Berkeley parser (Petrov and Klein 2007) that proposes the probabilistic context-free grammar with latent annotations. They reported a 73.92% F_1 score with the Stanford parser, 75.41% with the Bikel parser and 77.29% with the Berkeley parser based on a 10-fold cross-validation. These parsers are the state-of-the-art statistical parsers at the moment, in which parsers can elaborate context-free grammars by refining annotations of non-terminal symbols in the treebank. On the same extension, Park *et al.* (2016) continued to work on constituent parsing using the Sejong treebank. They obtained parsing results up to a 85.12% F_1 score (with gold part-of-speech (POS) labels) using the PCFG-LA Berkeley parser by correcting syntactic and POS annotation errors in the Sejong treebank. Note that Choi *et al.* (2012) and Park *et al.* (2016) used the black box experiments using the default option of each parser.

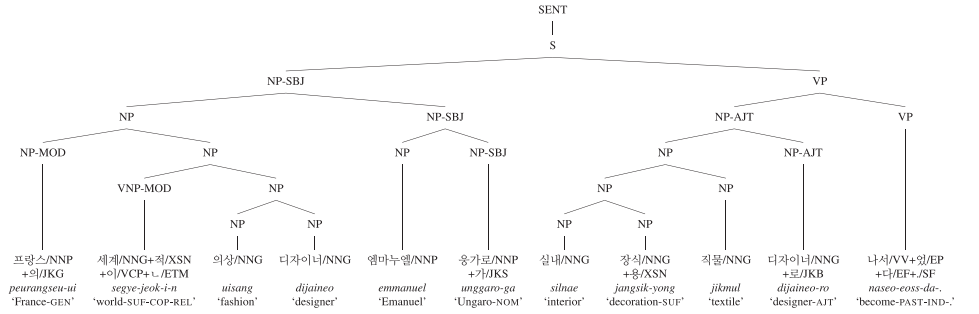
Seddah *et al.* (2013) and Seddah *et al.* (2014) presented the *modified* KAIST treebank at the SPMRL 2013 and 2014 shared tasks for Korean parsing. One of the participants obtained results up to 83.81% F_1 score for Korean by adding the reranking step to the Berkeley parser (Björkelund *et al.* 2013, 2014), in which reranking has been proven effective to refine parsing results (Collins 2000; Charniak and Johnson 2005; Collins and Koo 2015). While this result for Korean outperformed the earlier neural parsers such as Durrett and Klein (2015), more recent neural parsers reported up to 88.96% (Coavoux and Crabbé 2016; Kitaev and Klein 2018; Kitaev, Cao, and Klein 2019) from the same data set.

https://doi.org/10.1017/S1351324920000479 Published online by Cambridge University Press

(a)

```
; 프랑스의 세계적인 의상 디자이너 엠마누엘 웅가로가 실내 장식용 직물 디자이너로 나섰다.
(S (NP_SBJ (NP (NP_MOD 프랑스/NNP + 의/JKG)
               (NP (VNP_MOD 세계/NNG + 적/XSN + 이/VCP + ㄴ/ETM)
                   (NP (NP 의상/NNG)
                       (NP 디자이너/NNG))))
           (NP_SBJ (NP 엠마누엘/NNP)
                   (NP_SBJ 웅가로/NNP + 가/JKS)))
   (VP (NP_AJT (NP (NP (NP 실내/NNG)
                       (NP 장식/NNG + 용/XSN))
                   (NP 직물/NNG))
               (NP_AJT 디자이너/NNG + 로/JKB))
       (VP 나서/VV + 었/EP + 다/EF + ./SF)))
```

(b)



(c)

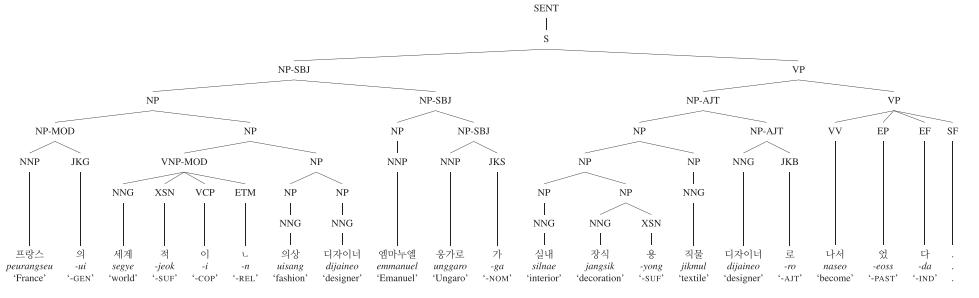


Figure 2. An example of the Sejong treebank. (a) The original Sejong treebank annotation. (b) The original Sejong treebank. (c) The Sejong treebank converted to the morpheme-based treebank for constituent parsing.

Hermjakob (2000), Sarkar and Han (2002) and the SPMRL KAIST data set utilised an English-like tokenisation scheme for the input, in which punctuation marks are separated from words.[b] However, many downstream applications for Korean language processing are based on morphological segmentation granularity, in which all morphemes are separated: POS tagging (Jung, Lee, and Hwang 2018; 2019), machine translation (Park *et al.* 2016, 2017), etc. The main reason they introduce this morpheme-based granularity is to avoid the data sparse problem because longer segmentation granularity such as eojeols can yield a number of possible words exponentially, and it results in the inevitable data sparsity problem. Therefore, current phrase structure parsing for Korean has also utilised morphemes-based granularity as a de facto standard (Chung *et al.* 2010; Choi *et al.* 2012; Park *et al.* 2016).

Whereas the interpretation of a direct comparison of results between the different granularity levels has been considered as being difficult because their representation levels are different (e.g., the number of terminal lexical tokens is different), we empirically found that morpheme-based granularity outperforms the English-like tokenisation scheme for Korean language

[b]Note that as eojeols have been used as a basic analysis unit in many Korean corpora, especially the Sejong corpus, the tokenisation task is often ignored for Korean.

https://doi.org/10.1017/S1351324920000479 Published online by Cambridge University Press

Table 1. Phrase structure parsing result comparison by Park *et al*. (2016)

	5-fold	10-fold	80-10-10
Parser	84.90	84.83	84.34
Gold	85.88	85.75	85.12

processing applications including constituent parsing if they are fairly converted into the same granularity level.

1.3 Goal of the paper

The recent development of constituent parsing has focused on attention-enhanced neural models (Vinyals *et al*. 2015; Kuncoro *et al*. 2017; Kitaev and Klein 2018; Kitaev *et al*. 2019) and domain-independent *agnostic* models (Vinyals *et al*. 2015; Fried, Kitaev, and Klein 2019). Reranking (Charniak and Johnson 2005) has been tested for neural constituent parsing (Fried, Stern, and Klein 2017), and dynamic oracles for dependency parsing (Goldberg, Sartorio, and Satta 2014) have also been applied to constituent parsing (Fernández-González and Gómez-Rodríguez 2018; Fried and Klein 2018). Using the Berkeley parser (Petrov and Klein 2007), the Trance parser (Watanabe and Sumita 2015) and Berkeley neural parser (Kitaev and Klein 2018; Kitaev *et al*. 2019), we train and evaluate the phrase structure Korean Sejong treebank. As shown in Dakota and Kübler (2017) for other languages and treebank, we define the best practices for replicability in constituent parsing for Korean, including correct comparison to future work by proposing the standard corpus division for the Sejong treebank. The other main contribution of this paper is to present two important factors on constituent parsing for Korean: detailed qualitative and quantitative parsing error analyses (Sections 3 and 4, respectively) using the Sejong treebank.

2. Experiments and results

2.1 Parsing evaluation methodology

In previous work on parsing for Korean, either phrase structure or dependency parsing, Park *et al*. (2013) proposed the 80-10-10 corpus split for training, development and evaluation, whereas others often used cross-validation (Choi *et al*. 2012; Oh and Cha 2013). Table 1 from Park *et al*. (2016) presents the current baseline parsing results using phrase structure grammars by the PCFG-LA Berkeley parser. Park *et al*. (2016) performed 5-fold and 10-fold cross-validation as well as corpus split evaluation methodologies for comparison purposes. Reported results are improved compared to Choi *et al*. (2012) because syntactic and POS tagging errors have been corrected in the Sejong treebank. Since results between different evaluation methods are relatively small, we propose to divide the corpus into a 80-10-10 ratio for the Sejong treebank.

2.2 Data and parsing systems

The current available version of the Sejong treebank (released in 2011) contains about 41,000 sentences in 15 files. The KAIST treebank used in SPMRL and the Penn Korean treebank in Chung *et al*. (2010) also used an 80-10-10 ratio for each file. While Choi and Palmer (2011) proposed the similar approach for dependency parsing, using the Sejong treebank, their version of the Sejong treebank is not currently available and they did not specify the exact file split. The contents of the Sejong treebank represent a variety of sources, newswire text, magazine articles on various subjects and topics, several book excerpts, and crawled texts from the internet. Therefore, we divide up each treebank file into an 80-10-10 ratio as in the previous work and merges them together into a training, a development and a test data set for constituent parsing. Currently,

https://doi.org/10.1017/S1351324920000479 Published online by Cambridge University Press

Table 2. The proposed number of sentences from the Sejong treebank to consist train/dev/test partitions

	train	dev	test	(total)
BGAA0001	1191	148	148	1487
BGAA0164	447	55	55	557
BGAE0200	3637	454	454	4545
BGBZ0073	3530	441	441	4412
BGEO0077	3572	446	446	4464
BGEO0292	3606	450	450	4506
BGEO0320	2844	355	355	3554
BGGO0098	2358	294	294	2946
BGGO0358	2096	261	261	2618
BGHO0107	761	95	95	951
BGHO0127	1193	148	148	1489
BGHO0409	1265	158	158	1581
BGHO0411	1427	178	178	1783
BGHO0431	1881	235	235	2351
BGHO0437	3416	426	426	4268
Total	33,224	4144	4144	41,512

it contains 33,224/4144/4144 sentences for training, development and test, respectively. Table 2 shows the proposed number of sentences in each file in the Sejong treebank. If appropriate, the proposed split can be served as a 90-10 ratio without a development data set.

For our experiments, we make use of the morphological analysis annotations in the treebank and focus on the parsing task, in which we directly use morpheme-separated information in the treebank instead of end-to-end parsing. For example, a relative clause verb *segyejeokin* ('world-class') is analysed and segmented into morphemes *segye* (NOUN), *-jeok* (DERIVATIONAL SUFFIX), *-i* (COPULAR) and *-n* (RELATIVE MARKER) in the treebank. POS tags are integrated as leaf nodes alongside their morphemes. We preprocess the treebank data by converting POS tags into immediate non-terminal labels as in the previous work by Choi *et al.* (2012). Consequently, the originally binary tree-structured Sejong treebank may contain multifurcating branches for the word such as [VNP-MOD [NNG *segye*] [XSN *jeok*] [VCP *i*] [ETM *n*]], and a word *segyejeogin* is converted into *segye* ⊔ *jeog* ⊔ *i* ⊔ *n* where ⊔ represents a blank space for the input of parsing systems.

Figure 2 shows an example of the Sejong treebank for the sentence in (1): Figure 2(a) for the original treebank annotation, Figure 2(b) for the tree-style original treebank and Figure 2(c) for the converted multifurcating treebank for constituent parsing.

We use the Berkeley parser (Petrov and Klein 2007)[c] for statistical parsing, and the Trance parser (Watanabe and Sumita 2015)[d] and the Berkeley neural parser (Kitaev *et al.* 2019)[e] for neural parsing with their default options. We invite you to see their `github` pages for the detailed

[c] https://github.com/slavpetrov/berkeleyparser.
[d] https://github.com/tarowatanabe/trance.
[e] https://github.com/nikitakit/self-attentive-parser.

https://doi.org/10.1017/S1351324920000479 Published online by Cambridge University Press

Table 3. Overall parsing results using PCFG-LA BERKELEY (Petrov and Klein 2007), TRANCE (Watanabe and Sumita 2015) and NEURAL BERKELEY (Kitaev and Klein 2018; Kitaev *et al*. 2019)

	PCFG-LA	P. BERKELEY		NEURAL	N. BERKELEY
	BERKELEY	+ POS GOLD	TRANCE	BERKELEY	+ BERT
w/o `funct`	84.31	85.47	87.24	88.67	89.18
w `funct`	82.64	83.47	86.22	86.75	88.20

option values. The Berkeley parser uses the probabilistic context-free grammar with latent annotations (PCFG-LA, Matsuzaki, Miyao, and Tsujii 2005) framework and performs series of split and merge cycles of non-terminal nodes to maximise the likelihood of a treebank. The Trance parser is a transition-based shift-reduce parser using compositional vector grammars described in Socher *et al*. (2013). The Berkeley neural parser uses a sentence encoder based on self-attention for a chart decoder (Kitaev and Klein 2018), and later extends to unsupervised pre-training using Bidirectional Encoder Representations from Transformers (BERT) (Devlin *et al*. 2019). The Berkeley neural parser reported that it outperformed the best result of SPMRL 2014 by Björkelund *et al*. (2014) for Korean. These parsers show competent results across a variety of languages, and they are easy to train on any treebank. These factors make the current experiments more objective and reproducible. The Trance parser can take word embeddings (Mikolov *et al*. 2013a; Pennington, Socher, and Manning 2014; Bojanowski *et al*. 2017), which has shown to benefit many natural language processing applications including the constituent parsing system such as Socher *et al*. (2013). We collect text from the Sejong raw corpus, the Korean Wikipedia (accessed on August 20, 2017), and all published articles during 2016 in *The Hankyoreh* ('The Korean Nation') for a 100-dimensional skip-gram model (Mikolov *et al*. 2013b) to build a word embedding vector. We use `bert-base-multilingual-uncased` for pre-training to combine with the Berkeley neural parser. For experiments in this paper and error analyses in Sections 3 and 4, we use Penn-like preprocessing for the treebank, especially by removing function labels in the phrase label such as `-sbj` or `-obj` as described in Bikel (2004b). We also convert the POS label `np` (pronoun) into `prop` to avoid confusing the noun phrase (also `np`) in the treebank.

2.3 Results

Table 3 shows overall parsing results using the Berkeley parser (Petrov and Klein 2007), the Trance parser (Watanabe and Sumita 2015) and the Berkeley neural parser (Kitaev and Klein 2018; Kitaev *et al*. 2019). We evaluate the parser accuracy with the standard F_1 metric from `EVALB`.[f] Park *et al*. (2016) reported that results from the Berkeley parser improve up to 85.12% using the gold POS tags. It is also feasible to use a 90-10 ratio without development data for the Berkeley parser. While the original Berkeley parser paper (Petrov *et al*. 2006) proposed several runs of training because of the EM algorithm which can find locally maximum likelihood parameters, we empirically found that each run of training gives the same results. Therefore, we can use train and dev data together for the single run of training using the PCFG-LA Berkeley parser. While we obtain a 84.31% F_1 score without gold POS labels and 85.47% with gold POS labels for a 90-10 corpus split, the difference of results are still relatively small.[g] We obtain 87.24% using the Trance parser.[h] We train and evaluate the Berkeley neural parser (Kitaev and Klein 2018; Kitaev *et al*. 2019). It obtains up to 88.67% and 89.18% f_1 scores without and with BERT, respectively. The result by the Berkeley

[f]http://nlp.cs.nyu.edu/evalb. There is also a modified version, `EVALB_SPMRL` released with the SPMRL shared task, http://www.spmrl.org. There are no differences for the current reported results using both evaluation scripts.

[g]The parsing model for the Berkeley parser to reproduce the result is available at http://doi.org/10.5281/zenodo.3995084.

[h]The parsing model including the pre-trained 100-dimensional skip-gram word embeddings for the Trance parser to reproduce the result is available at http://doi.org/10.5281/zenodo.1004604.

https://doi.org/10.1017/S1351324920000479 Published online by Cambridge University Press

Table 4. Parsing results using PCFG-LA BERKELEY per iteration with the numbers of non-terminal nodes and CFG rules

	w/o funct			w funct		
Iteration	# of NT	# of CFGs	F_1	# of NT	# of CFGs	F_1
1	98	3335	69.07	209	10,130	70.02
2	146	10,448	72.38	314	31,158	74.93
3	218	32,039	78.75	472	91,956	79.15
4	326	97,224	81.62	711	255,381	81.28
5	488	297,487	83.47	1070	715,264	82.29
6	731	812,790	84.31	1608	2,027,339	82.64

neural parser using BERT (Kitaev *et al.* 2019) significantly outperforms other previous parsing results using the Sejong treebank.[i] The observed *p*-values compared to results by the Berkeley parser (Petrov and Klein 2007) and the Trance parser (Watanabe and Sumita 2015) are much less than 5%.[j]

Chung *et al.* (2010) reported that including function labels such as `-sbj` or `-obj` can improve parsing results for the Penn Korean treebank. While such function labels appear only at `np` in the Penn treebank, any phrase labels can have function labels in the Sejong treebank. For example, *miggeuleoji* ('slide', a verb), *-m* (NOUN DERIVATIONAL SUFFIX), and *-do* ('also') is annotated as `vp-sbj` in the Sejong treebank because a noun derived from a verb is annotated as a `vp` and its function in the sentence is a subject. Such diversities of phrase labels with function labels in the Sejong treebank can give a biased distribution of phrase labels, and we hardly improve parsing results using such labels. Practically, the result with function labels is decreased by between 1% and 2% for all statistical and neural parsers. Table 4 shows parsing results using the Berkeley parser (Petrov and Klein 2007) per iteration with the numbers of non-terminal nodes and CFG rules. As the number of non-terminal nodes increases, the number of the rules of CFG increases exponentially, thereby leading to the increase in the complexity in parsing. Taking this result into consideration, this study can make a judgement that using such a phrase label would bring a certain degree of limitations to improving the practical effect of their results. The same is true of the French treebank as described in Park (2018). The improvement in the results of Chung *et al.* (2010) using the Penn Korean treebank might be due to the restrictions corresponding to the relatively small number of sentences in the treebank.

3. Qualitative parsing error analysis

In this section, we find six different types of common errors in the course of qualitatively analysing the parsing data entirely based on our linguistic intuitions.[k]

1. error from dropping (case) markers,
2. error in recognising arguments of a predicate,
3. error in modifying verb phrases,
4. error in modifying the noun phrase (NP) with relative clauses,

[i]The parsing models for the Berkeley neural parser including detailed hyperparameters to reproduce the result are available at http://doi.org/10.5281/zenodo.3995068.

[j]The *p*-values are calculated using Bikel's script available at https://github.com/tdozat/Parser-v1/blob/master/bin/compare.pl.

[k]We use parsing results for qualitative error analysis using the PCFG-LA Berkeley parser proposed in Park *et al.* (2016). The parsing model and the pipeline script for the Berkeley parser are available at http://doi.org/10.5281/zenodo.891267.

https://doi.org/10.1017/S1351324920000479 Published online by Cambridge University Press

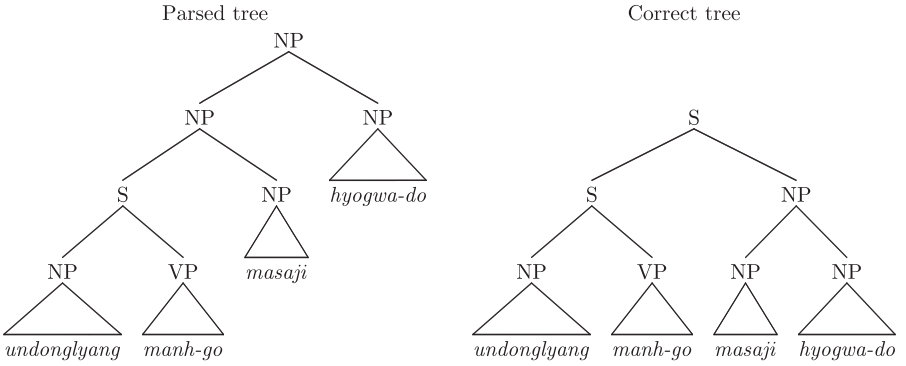


Figure 3. Error from dropping (case) markers: the first noun phrase *undonglyang* ('amount of exercise') should have a nominative case marker *-i* as in *undonglyang-i*.

5. error in identifying the category, and
6. error in identifying lexical properties like polyseme and homonym.

Most of these types of errors are mainly due to the linguistic characteristics of the Korean language such as head-final word order, and case, pronoun and conjunction dropping behaviours caused by the morphological richness. The Korean language has SOV (subject-object-verb) word order, which means that the head in a sentence occurs in a final position and that a sentence can, at least, take two consecutive NPs and that the distance between the main verb and its subject can also be much bigger than we might expect. These linguistic properties of the Korean language significantly influence the following types of errors in parsing. We detail these errors, and the following figures present parsed results and correct parse trees.

3.1 Error from dropping (case) markers

Among the linguistic features of the Korean language stated above, the three linguistic features of dropped case marker, pronoun and coordinating conjunction trigger the errors in appropriately modifying NPs. In the Korean language, case markers attached to NPs can be omitted. They would be both markers such as nominative and accusative cases in the subject or object positions, genitive case of nouns preceded by the other nouns, and the postpostional markers attached to nouns denoting place, direction or time, etc. The genitive case marker, of course, might be found wherever two consecutive NPs occur in a sentence. The following example in (2) (as described in Figure 3) lies in the first case, the drop of a nominative case marker in subject position. The first NP *undonglyang* ('amount of exercise') should have a nominative case marker *-i* as in *undonglyang-i*, but it was omitted. This resulted in a parsing error. That is, it leaves the possibility in which the predicate *manh-go* can easily form one constituent with the immediate following element *masaji* ('massage'), in particular because this predicate is connected to the coordinate conjunction *-go* and it usually tends to make a constituent with the following NP *masaji*, which, in fact, functions without a genitive marker as a modifier of the NP *hyogwa* ('effect'). After all, it results in an erroneous analysis, where the NP *masaji* does not modify the immediate NP *hyogwa*. If the drop of the nominative case marker did not happen, the subject NP would be easily combined with the constituent *manh-ko* immediately followed by.

(2) 운동량 많고 마사지 효과도 . . .
undonglyang *manh-go* *masaji* *hyogwa-do*
amount of exercise-nom being high-CONJ massage-gen effect-also
'The amount of exercise is high and the effect of massage also. . .'

https://doi.org/10.1017/S1351324920000479 Published online by Cambridge University Press

Table 5. Errors from argument dropping of a predicate: incorrect match of subject or object in embedded clause by matrix verb

Position	Dropped elements
Subject NP	Subject (in embedded clause)
Object NP	Object (in embedded clause)

3.2 Error in identifying arguments of a predicate

Error in identifying the arguments of a predicate – a subject or sometimes an object – can be found in two situations, as shown in Table 5. One is the absence of subject in an embedded clause and the other is a rather long distance between the verb and its subject. First of all, the Korean language typically is a pro-drop language in which some pronouns may be omitted when they are contextually inferable. This property causes errors in identifying the subject when an embedded or matrix verb looks for its subject. Sometimes, the object itself can also lead to the same errors. Thus, when the pronoun in subject position is omitted, particularly in an embedded sentence, the verb in a subordinate clause can wrongly recognise the subject of a main clause as its own subject, since the subject in a subordinate clause has a relatively high possibility of being deleted if it has the same referent to that in the main clause. In addition to the pro-drop property, the Korean language displays an SOV word order, which means that the head occurs in the final position of the sentence and the distance between the main verb and its subject may be long. This linguistic property also leads to errors. Sometimes the pro-drop subject and the long distance between it and the verb combine to mislead the parsing.

The following sentence (3) (Figure 4) consists of two clauses, one of which has an embedded clause without a subject pronoun *angyeong-eul sseosseul ttae* ('eyeglass-ACC wear'). This environment causes error in recognising its subject. In this clause, the verb *sseosseul* ('wear') wrongly identified the subject of the main clause *hangug sobija bohowon-i* ('Korea Consumer Agency-NOM) as its own subject. The subject in the subordinate clause is not presented but, nonetheless, we can easily understand from the context that the entity who does the action of wearing eyeglasses would be people in general. That is, no one would comprehend that the actor wearing them must be the subject of a main clause, *hangug sobija bohowon*.

(3) 한국 소비자 보호원이. . . 성능을 조사한 결과
hangug sobija bohowon-i. . . seongneung-eul josaha-n gyeolgwa,
Korea Consumer Agency-NOM. . . performance-ACC inspect result,
안경을 썼을 때. . . 제품이 27.5% 나 되는
angyeong-eul sseosseul ttae. . . jepum-i 27.5% na doeneun
glass-ACC wear moment. . . products-NOM 27.5% around was
것으로 나타났다
geos-eulo natana-ss-da
that-by be-shown
'As the survey result of Korea Consumer Agency, it was shown that. . . products are around 27.5% when one wears eyeglasses'

3.3 Error in modifying VP

When adverbs modify verb phrases (VPs), parsing errors become more likely. Especially, under the circumstance where the main verb in a sentence containing more than one embedded clause is modified by adverbial expressions denoting time and location, the distance between them can be longer than that of the verb in the embedded clause. A situation such as a long dependency between two related elements can presumably lead to the errors in modifying VP. In particular,

https://doi.org/10.1017/S1351324920000479 Published online by Cambridge University Press

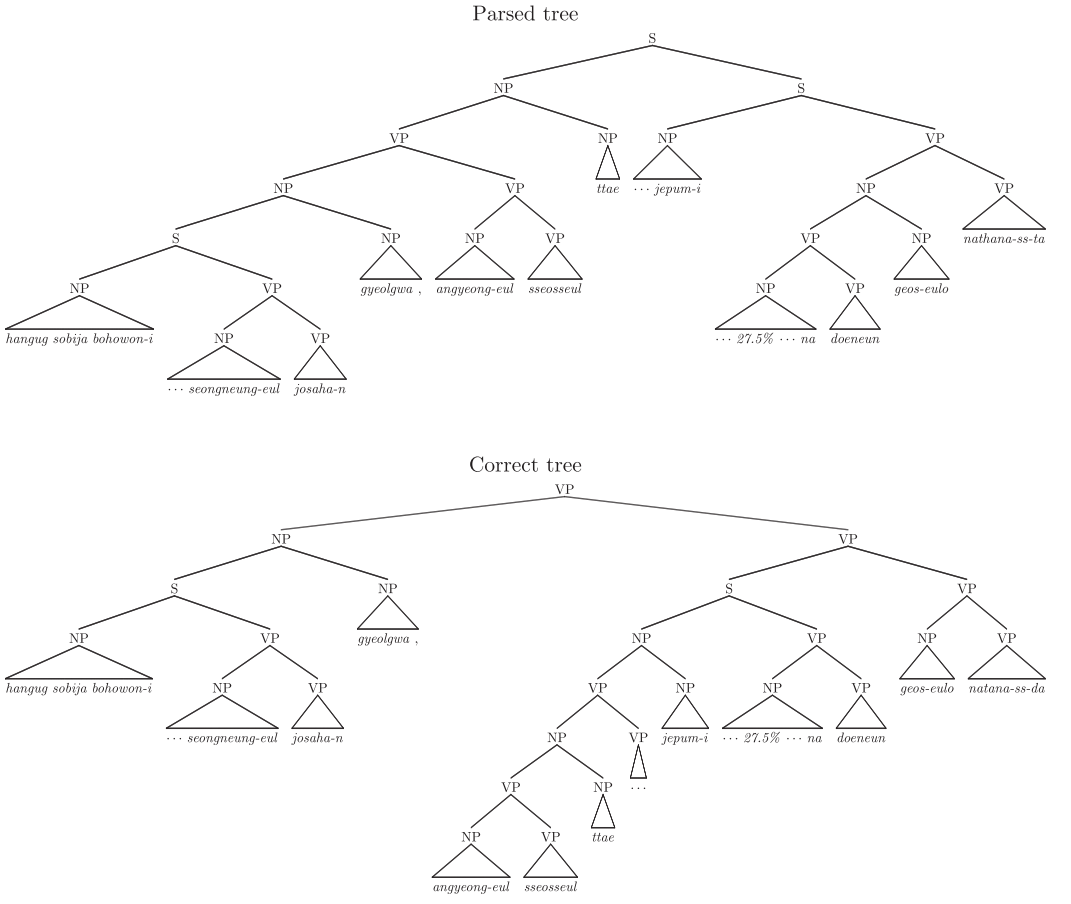


Figure 4. Error in identifying arguments of a predicate: the verb *sseosseul* ('wear') wrongly identified the subject of the main clause *hangug sobija bohowon-i* ('Korea Consumer Agency-NOM) as its own subject.

when the adverbial expressions occur in the initial position of a sentence, we can quickly imagine two possible alternatives to verb modification: to modify the closest verb or the main verb in the final position. That is, sometimes, the adverbial expressions can modify the embedded verb occurring in the closer position than the main verb happening in the final position of the whole sentence. Sometimes, the adverbs of sentences without long dependency relation can also wrongly modify the verb in the final position by crossing beyond the element necessary to be correctly modified from them. For instance, the adverbial expression in the following sentence *82 nyeon-eneun* ('82 year-in') should modify the immediately following verb *yeolly-eoss-deon* ('held when'), but it leads to an error by modifying the main verb in the final position of the sentence. See the example in (4) (Figure 5).

(4) 82년에는 한차례 열렸던 . . . 대회가
82-nyeon-eneun hanchalye yeolly-eoss-deon . . . daehoe-ga
82-year-in once held-when . . . competetion-NOM
92년 한 해 동안은 8개나 열릴 정도로 . . .
92-nyeon han hae dongan-eun 8-gaena yeollil jeongdo-lo . . .
92-year one year during-TOP 8 times held as like . . .

https://doi.org/10.1017/S1351324920000479 Published online by Cambridge University Press

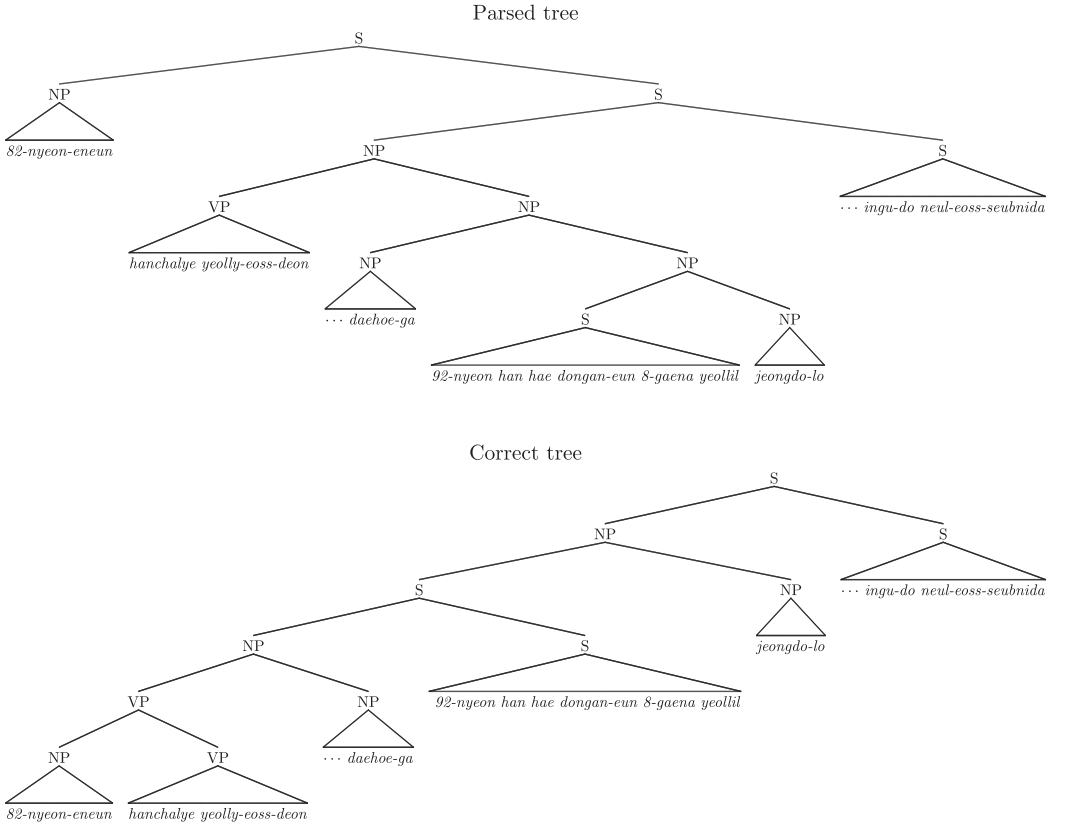


Figure 5. Error in modifying VP: *82 nyeon-eneun* ('82 year-in') should modify the immediately following verb *yeolly-eoss-deon* ('held when').

인구도 늘었습니다
ingu-do *neul-eoss-seubnida*
participants-also increased
'the competition which was held only one time in 1982, took place 8 times in 1992 and the participants were also increased'.

3.4 Error in modifying the NP with RC

NPs can typically be modified by relative clauses, but that can give rise to errors. When a relative clause modifies a NP containing more than one NP, the relative clause can modify inappropriate nouns. For instance, it only modifies the first noun instead of the second or last one, generating errors by forming a wrong constituent. The following sentence has a relative clause *deowi-leul sighi-neun* ('heat-ACC cool off-AJT'), and this modifies the head noun with two NPs *yeoleum-yong* ('use for summer'). This situation produces an error in wrongly modifying the first head noun, forming a wrong constituent because the second NP in head nouns of a relative clause (RC) is regarded as another constituent and only the first NP is modified by the relative clause. See the example in (5) (Figure 6).

(5) 더위를 식히는 여름용과는 달리 집안
deowi-leul *sighi-neun* *yeoleum-yong gwaneun* *dalli* *jiban*
heat-ACC cooling off use for summer unlike in the house

https://doi.org/10.1017/S1351324920000479 Published online by Cambridge University Press

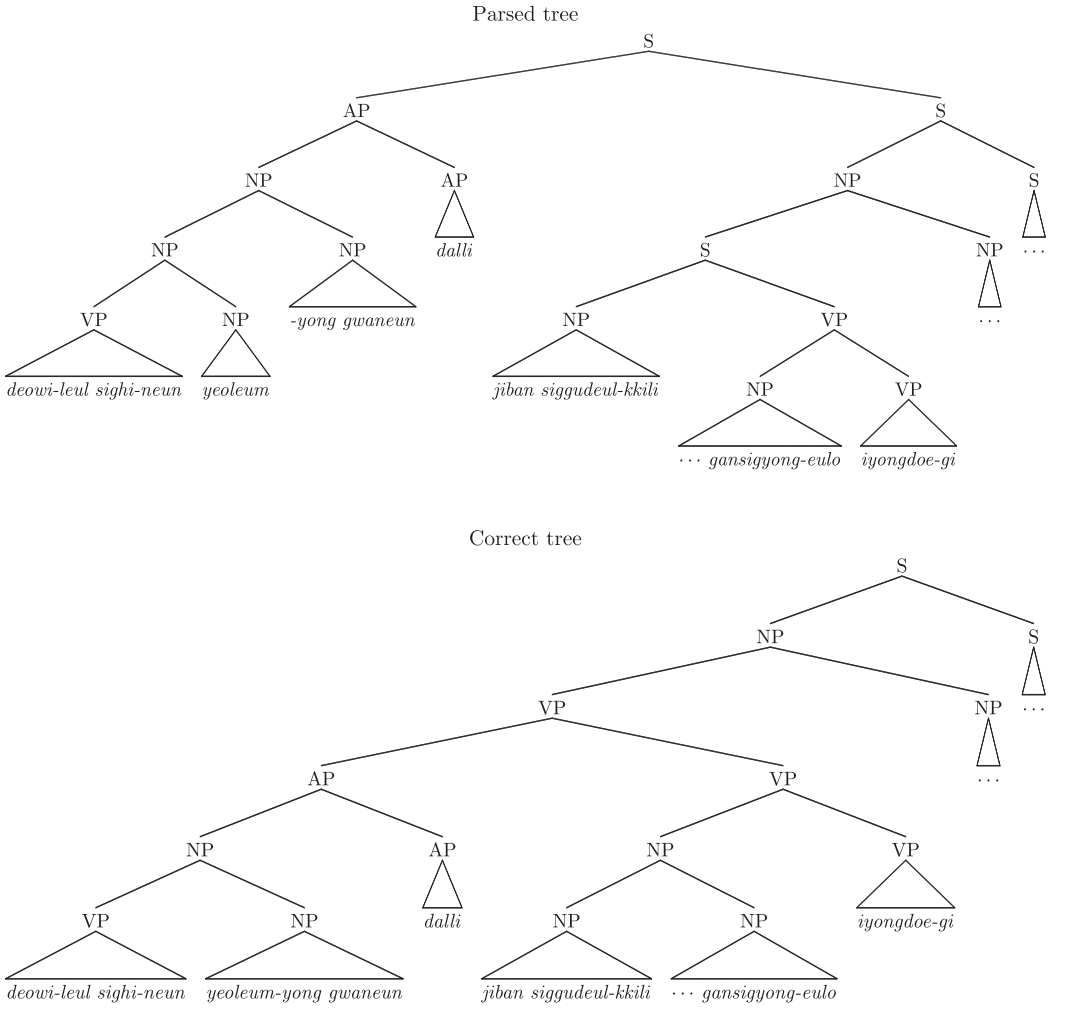


Figure 6. Error in modifying the noun phrase with RC: *deowi-leul sighi-neun* ('heat-ACC cool off-AJT') should modify the head noun with two noun phrases *yeoleum-yong* ('use for summer').

식구들끼리 ...간식용으로 이용되기
siggudeul-kkili ... gansigyong-eulo iyongdoe-gi
for only family ... light meals-for to be used
'Rather than as summery desserts to cool off the summer heat, it would be used for only family as light meals'

3.5 Error in identifying category

As stated at the beginning of this section, the Korean language features a case marker drop. The contracted form of bound morphemes can also trigger errors in identifying phrasal categories in certain phrasal levels. The phrase *anbang-seo* in the following sentence (6) has the contracted form *seo* of bound morpheme *e-seo*. If we recover the source sentence of the following sentence, the postpositional marker *e-seo* (or *seo*, LOC) can be attached to the noun *anbang* ('main room'), resulting in forming a prepositional phrase denoting a location. Unfortunately, this contracted form *seo* has two meanings in this context: as a postpositional marker denoting a location or as a coordinating conjunction for connecting two events. In the sentence below, the bound morpheme

https://doi.org/10.1017/S1351324920000479 Published online by Cambridge University Press

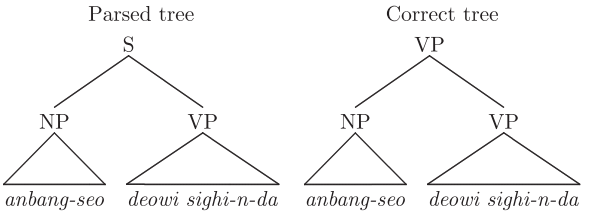


Figure 7. Error in identifying category: the bound morpheme *seo* has to be used as a postpositional marker denoting a location.

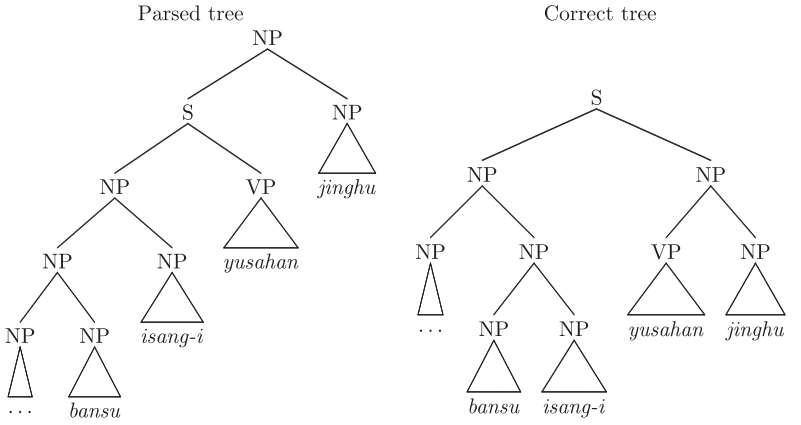


Figure 8. Error by lexical properties like polysemy and homonym: *isang*, should be interpreted as a meaning of 'more than'.

seo has to be used as the former, but it was parsed as the latter, ending up with an error in recognising the wrong phrasal category for the sentence. See the example in (6) (Figure 7). We limit this type of error only to the case where detecting phrase boundaries is correct and the labelling of syntactic tags is not correct.

(6) 안방서 더위를 식힌다
anbang-seo *deowi* *sighin-da*
main room-LOC summer-heat cool off
'We cool off in the main room'.

3.6 Error by lexical properties like polysemy and homonym

Some lexical items with multiple meanings or with the same spelling but different meanings also lead to parsing errors. In the following example, we can find *isang*, which can be used as a polysemous word. This lexical word can be interpreted simultaneously as meaning 'ideal' as well as 'more than'. In the former, this word can be a new single phrasal category by combining with the immediately preceding word, whereas in the latter, it denotes the amount of the preceding noun and functions as a quantifier. This lexical property of polysemy yields to errors in forming a proper constituent. In the context of the sentence in (7) (Figure 8), this word was used as the latter.

(7) 반수 이상이 유사한 징후
bansu isang-i *yusahan jinghu*
half more than-NOM similar symptom
'more than half is shown as similar symptom'

https://doi.org/10.1017/S1351324920000479 Published online by Cambridge University Press

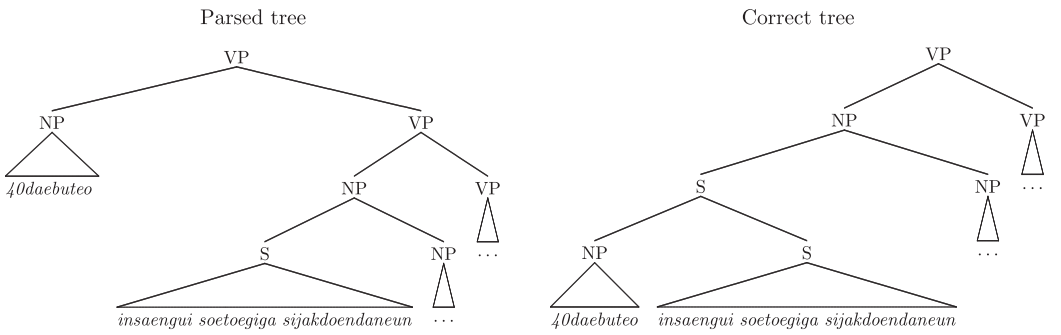


Figure 9. NP attachment error: the noun phrase *40daebuteo* ('from 40s') should be the argument of the subordinate clause *insaengui soetoegiga sijakdoendaneun* ('has started the decline of life').

3.7 Discussion on qualitative error analysis

We explored qualitative error analysis by focusing on phrase boundary detection errors which yield incorrect boundaries and labels. The morphological richness and even SOV word order can also significantly influence the types of errors in parsing. We believe that current qualitative analysis might be estimated to cover over 90% of the parsed sentences with errors as measured by thoroughly examining manually the parsing results with our linguistic intuitions. There are also some minor errors such as `advp advp` errors for the adverbial phrase, in which a sentence is parsed either [$_S$ [$_{ADVP}$ [$_{ADVP}$ *adverb*] [$_{ADVP}$ *adverb*]] [$_S$ *sentence*]] or [$_S$ [$_{ADVP}$ *adverb*] [$_S$ [$_{ADVP}$ *adverb*] [$_S$ *sentence*]]] because the Sejong treebank annotates the sequence of the adverbial phrases in both ways inconsistently. It would be, however, difficult to quantify parsing errors that result from qualitative classification, for example, the number of errors from dropping case markers where we cannot count the number of dropping case markers. In addition, a distinction between errors might remain vague, for example, an error in Section 4.2 can be classified as an error in recognising arguments of a predicate and an error in modifying the NP with relative clauses. Therefore, we develop an error typology based on phrase labels in the next section to quantify parsing errors, from which we may systematically elaborate to improve future parsing results.

4. Quantitative parsing error analysis

4.1 Methodology

Kummerfeld *et al.* (2012, 2013) presented an approach to quantify constituent parsing errors for English and Chinese. We extend this method to Korean by adding `advp` attachment and word boundary errors. The following errors are considered from parsing results. We use parsing example for quantitative error analysis using the PCFG-LA Berkeley parser.

4.2 Quantitative error typology

NP attachment NP attachment errors are the most frequent errors in constituent parsing for Korean. They consist in mistakes and inconsistencies in recognising arguments of a predicate. While the first NP *40dae-buteo* ('from 40s') in (8) (Figure 9) is an argument of the subordinate clause *insaeng-ui soetoegi-ga sijakdoe-n-da-neun* ('has started the decline of life'), it is dominated by the main clause and it results in an erroneous analysis.

(8) 40대 부터 인생의 쇠퇴기가 시작된다는
40dae buteo insaeng-ui soetoegi-ga sijakdoe-n-da-neun . . .
40s from life-GEN decline-NOM start-PRES-IND-REL . . .
' . . . has started the decline of life from 40s . . . '

https://doi.org/10.1017/S1351324920000479 Published online by Cambridge University Press

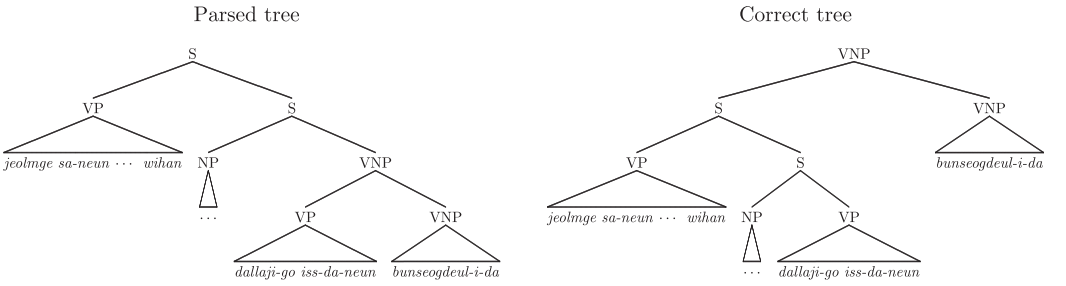


Figure 10. VP attachment error: the verbal phrase *dallaji-go-i-ss-da-neun* ('being differentiated') should be the predicate of the subordinate clause. It also shows a different label error where the root non-terminal symbol should be a VNP.

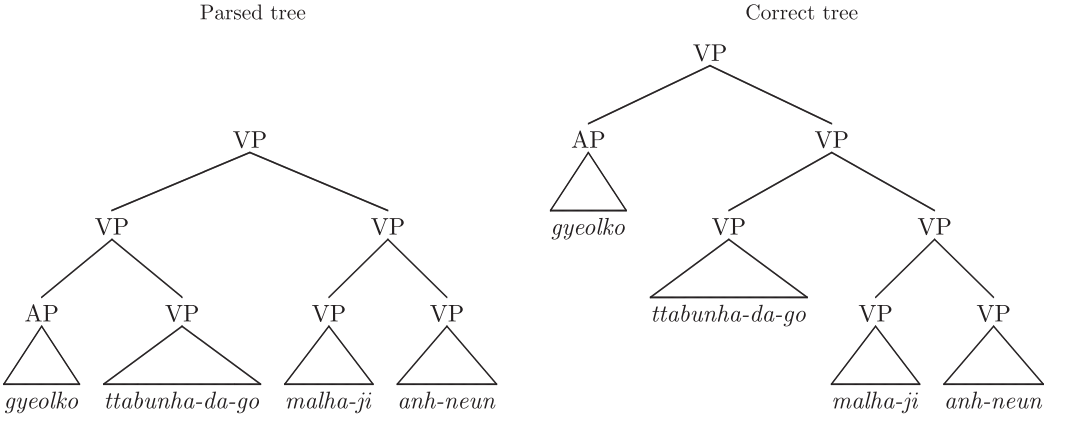


Figure 11. Adverbial phrases attachment error: the AdvP *gyeolko* ('never') should modify the whole sentence *malhaji anhneun* ('not say).

VP attachment When two (or several) verbal phrases (VPs) appear consecutively and a parser identifies them as complex predicates, it results in (9a) (Figure 10). However, it leads to an error by modifying the main verb in the final position of the sentence. A VP *dallaji-go-i-ss-da-neun* ('being differentiated') is the predicate of the subordinate clause and the entire subordinate clause should modify the main verb *bunseogdeul-i-da* ('be analyzed') in (10) and Figure 10.

(9) 달라지고 있다는 분석들이다
dallaji-go *iss-da-neun* *bunseogdeul-i-da*
differentiate-CONJ be-IND-REL analyses-COP-IND
'it is analyzed that . . . being differentiated . . . '

AdvP attachment When adverbial phrases (AdvPs, annotated as an AP in the Sejong treebank) modify verb phrases, parsing errors become more likely. Adverbs and adverbial phrases modifying whole sentences often appear at the beginning of the sentence. While an AdvP *gyeolko* ('never') in (10) (Figure 11) modifies the whole sentence *malhaji anhneun* ('not say') at the beginning (of VNP), the parser analyses it as modifying the VP of the subordinate clause *ttabunhadago* ('being bored').

(10) 결코 따분하다고 말하지 않는
gyeolko *ttabunha-da-go* *malha-ji* *anh-neun*
never bored-IND-CONJ say not-REL
'. . . never say . . . being bored . . . '

https://doi.org/10.1017/S1351324920000479 Published online by Cambridge University Press

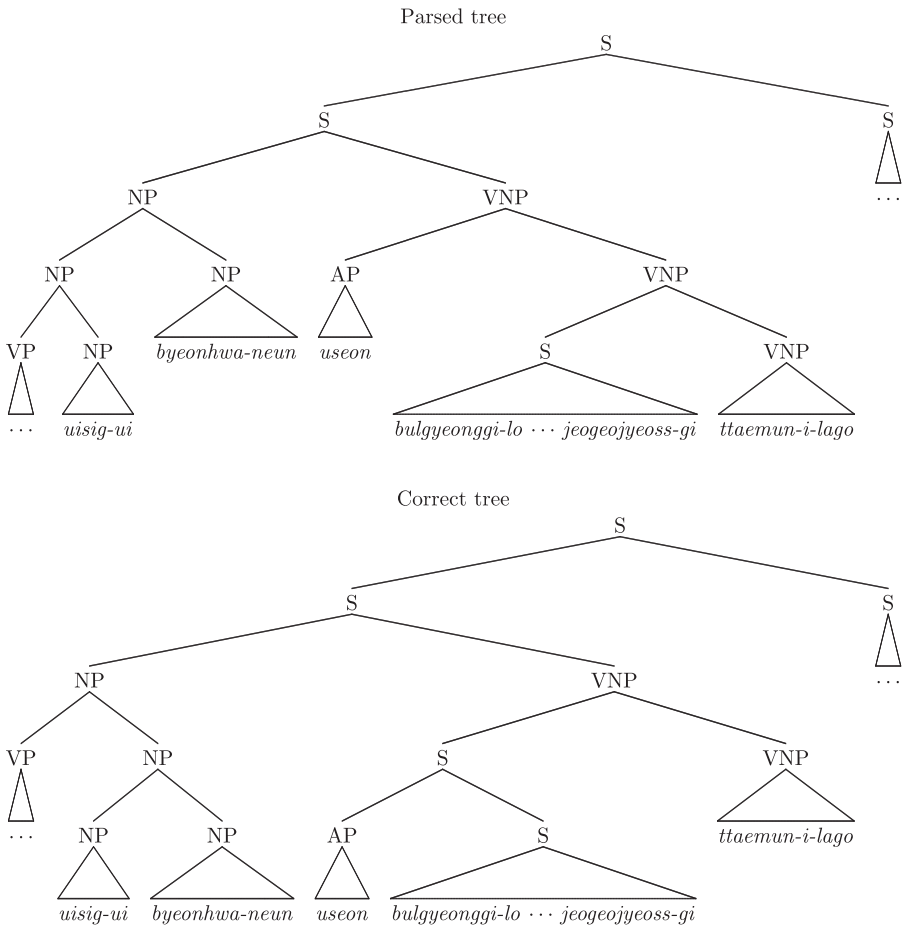


Figure 12. Clause attachment error: the predicate of the clausal argument *jeogeojyeoss-gi* ('being decreased') should be analysed as a complex predicate.

Clause attachment Clause attachment errors are similar to VP attachment errors. We differentiate them based on their phrase labels, either S or VP. The predicate of the clausal argument *jeogeojyeoss-gi* ('reducement') in (11) (Figure 12) is analysed as a complex predicate, in which it is dominated by VNP (noun verbal phrase) also because an AdvP *useon* ('first') is also wrongly parsed to modify the entire VNP.

(11) 우선 불경기로 . . . 적어졌기 때문이라고
useon bulgyeonggi-lo . . . jeogeoji-eoss-gi ttaemun-i-lago
first depression-AJT . . . reduce-PAST-SUFF caused-COP-QUOTE
' . . . because . . . first be reduced . . . in depression . . . '

Word boundary Compound words in Korean can be written alternatively with or without a blank such as *yeoreobunyaeseo* or *yeoreo bunyaeseo* ('in several fields'). Since *yeoreo* ('many') in (12) (Figure 13) also appears independently several times in the corpus, the parser analyses a single word *yeoreobunyaeseo* as *yeoreo bunyaeseo* because the input is morpheme-separated (*yeoreo bunya eseo*).

https://doi.org/10.1017/S1351324920000479 Published online by Cambridge University Press

Table 6. Quantitative error analysis results. The number of errors based on parsing results w/o function labels: NP for np attachment errors, WB for word boundary errors, VP for vp attachment errors, DL for different label errors, CL for clause attachment errors, ADVP for advp (adverbial phrase) attachment errors and UN for undefined errors

	PCFG-LA	P. BERKELEY		NEURAL	N. BERKELEY
	BERKELEY	+ POS GOLD	TRANCE	BERKELEY	+ BERT
NP	3946	3790	3373	3035	2875
WB	2792	2495	2282	2135	2113
VP	1648	1616	1382	1157	1158
DL	1340	1026	866	1870	848
CL	813	754	717	658	602
ADVP	293	284	227	213	220
UN	2728	2594	2218	2477	2433
Total	13,560	12,559	11,065	10,545	10,249

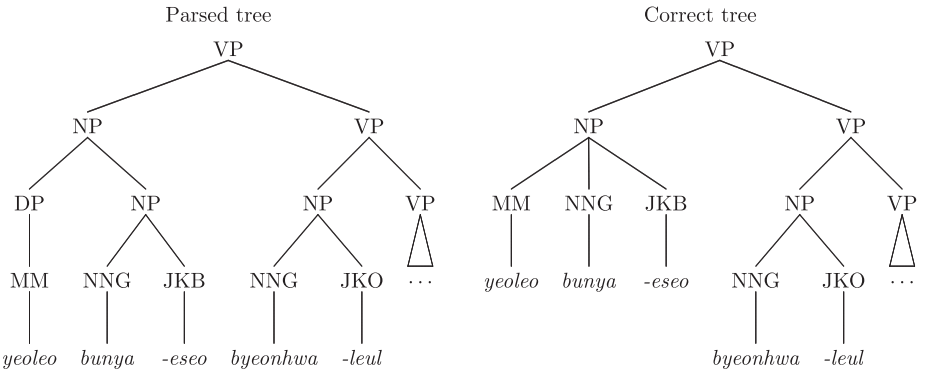


Figure 13. Word boundary error: the parser should analyse a single word *yeoreobunyaeseo*.

(12) 여러 분야에서
yeoleo bunya-eseo . . .
many fields-IN . . .
'in several fields . . . '

Different label Lastly, the parser can attribute wrong phrase labels such as S instead of VNP for sentences as in (9) (Figure 10) because there are sentences in which their starting symbols are a verbal phrase label (VP or VNP).

4.3 Discussion on quantitative analysis

Table 6 shows the numbers of errors in the test data set based on the different parser settings. Since the neural parsing system outperforms the statistical parsing systems for all types of errors, it would be easy to conclude that the recent neural parser performs better for the any sentence pattern qualitatively and quantitatively throughout the treebank. While Kummerfeld *et al.* (2012) presented different error quantities for English depending on parsers, the ratio for error types are

https://doi.org/10.1017/S1351324920000479 Published online by Cambridge University Press

consistent for Korean regardless of the parser type. Therefore, presented parsers show the similar tendency in the ratio of the frequency for error types. The best neural system yields up to less than 27.31% error rate for all attachment errors compared to the baseline PCFG-LA BERKELEY setting. It especially reduces the error rate by 36.71% for phrase labelling. We have 35,565 NPs, 28,304 VPs, 2976 AdvPs and 4010 S (clause) fragments in the test data set. It suggests that about 10% of NP (NP), AdvP (ADVP) and S (CL) labels are involved in parsing errors while less than 5% for VP (VP) labels for the PCFG-LA Berkeley parser. For example, we observe 11.0951% of NP fragments as NP attachment errors which would be closely related with errors from dropping case markers (Section 3.1) and from identifying arguments of a predicate (Section 3.2) in qualitative parsing error analysis. Therefore, reductions in NP errors by correctly identifying arguments of the predicate, for example, would result in large gains on F-scores. As Kummerfeld *et al*. (2012) remarked, some of the *undefined* errors could be categorised in errors described above; *undefined* errors can also contain a long tail of other errors, which we do not define. About 20% of parsing errors are undefined, and they require more fine-grained definition for quantitative error types, and we leave this for future work.

5. Technical discussion on parsing error analyses

As mentioned in earlier sections, this study mainly keeps close track of both the qualitative and quantitative analyses in order to find a good strategy for improving the results of constituent parsing for Korean. This study expects that this would lead to better results in the sense that two analyses can complement each other, by looking at their results with a multilateral perspective.

First of all, we could confirm that both analyses produce well-grounded results than we anticipated at the outset. The qualitative error analysis reliably allows us to detect phrase boundary errors, decisively influencing the error types in parsing triggered by morphological richness and SOV word order. The quantitative analysis tends to provide stable results regardless of the parser types.

The qualitative analysis, on the other hand, is rather sensitive to the methods of annotating the sequence of the phrases. This study suggests an error typology based on phrase labels as an alternative strategy to compensate for this difficulty, which is a quantitative analysis to make it possible to quantify parsing errors. The quantitative analysis, however, is likely to yield undefined parsing errors, thus requiring more refined definition for quantitative error types. As can easily be noticed by many researchers, it would be great to effectively use a new method that takes the interaction of the current qualitative and quantitative analyses into consideration. We have some difficulty in further looking at the errors in qualitative analysis by comparing them with the recent neural parsers similarly to that of the quantitative analysis in two different aspects.

One reason lies in that it is unlikely to be easy to keep consistent track of errors in both the two analyses because the same type of error can occur more than one time within the same or part of the sentence in the qualitative analysis, unlike that in the quantitative analysis which is in a deterministic way. Several errors, sometimes, might appear within the same sentence. In addition, the qualitative analysis in this study has been performed by thoroughly investigating manually the parsing results with our linguistic intuitions with an expert knowledge in the fields of Korean syntax. It is hard to find a more good method to perform the work automatically than ours, inevitably leaving this for future work. This analysis needs more researching in the field of constituent parsing for Korean in the near future. The new attempts, therefore, to complement the shortcomings of these two analyses would face a new challenge to overcome, as stating above.

6. Conclusion

Treebank grammars used to be readily understandable even in the various annotation schemes, such as markovisation and head lexicalisation (Charniak 1996; Johnson 1998; Klein and Manning

https://doi.org/10.1017/S1351324920000479 Published online by Cambridge University Press

2001; Post and Gildea 2009). Latent annotations in the Berkeley parser (Petrov *et al*. 2006; Petrov and Klein 2007), on the other hand, made it difficult for humans to interpret treebank grammars. Whereas there are efforts to understand recent neural language models (Kulmizev *et al*. 2020), the neural parsers, which estimate the parameters of the parsing model, came to be more difficult for humans to understand. Using the statistical and the neural parsers, however, this study has primarily tried at finding and identifying the concrete typology of parsing errors and additionally attempted to check how skillfully these parsers deal with sentences, focusing on the tendency of their distributional ratio. The most frequently occurring errors, no matter whether qualitative or quantitative analysis, concern the attachment problems mostly for the sentences with long distance dependency, in which the parser fails to find the correct argument of the predicate. Park *et al*. (2013) proposed a lexicalised probabilistic model with case frames aggregated from predicate-argument structures, and they improved the dependency parsing results for Korean. We expect to get some benefit from such information as case frames for constituent parsing, and we leave this for future work.

The main goal of the paper is to discuss widespread complicated issues on constituent parsing for Korean such as preprocessing of the treebank and error analysis, thereby improving parsing results. As a result of these works, we have properly defined where to train and evaluate the Sejong treebank, detailed constituent parsing errors, and have improved parsing results. We obtained a 89.18% F_1 score using the neural parser, and it outperformed other previous work for Korean using the Sejong treebank.

Over the last few years, we have worked on many extrinsic factors for constituent parsing to improve parsing results for Korean: Choi *et al*. (2012) transformed the treebank into entity level for a certain number of morphemes, and Park *et al*. (2016) and this paper adapted the recent neural parser for Korean parsing. Our future work will focus on a parsing scheme itself to adapt linguistic properties, such as the morphological richness and SOV word order, which significantly influence the parsing results for Korean.

References

Abeillé A. (ed.) (2003). *Treebanks: Building and Using Parsed Corpora*. Netherlands: Springer.

Bangalore S. and Joshi A.K. (1999). Supertagging: An approach to almost parsing. *Computational Linguistics* **25**(2), 237–265.

Bikel D.M. (2004a). Intricacies of Collins' parsing model. *Computational Linguistics* **30**(4), 479–511.

Bikel D.M. (2004b). *On the Parameter Space of Generative Lexicalized Statistical Parsing Models*. PhD Thesis, University of Pennsylvania.

Björkelund A., Çetinoğlu Ö., Faleńska A., Farkas R., Mueller T., Seeker W. and Szántó Z. (2014). Introducing the IMS-Wrocław-Szeged-CIS entry at the SPMRL 2014 shared task: Reranking and morpho-syntax meet unlabeled data. In *Proceedings of the First Joint Workshop on Statistical Parsing of Morphologically Rich Languages and Syntactic Analysis of Non-Canonical Languages*, Dublin, Ireland. Dublin City University, pp. 97–102.

Björkelund A., Çetinoğlu Ö., Farkas R., Mueller T. and Seeker W. (2013). (Re)ranking meets morphosyntax: State-of-the-art results from the SPMRL 2013 shared task. In *Proceedings of the Fourth Workshop on Statistical Parsing of Morphologically-Rich Languages*, Seattle, Washington, USA. Association for Computational Linguistics, pp. 135–145.

Bojanowski P., Grave E., Joulin A. and Mikolov T. (2017). Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics* **5**, 135–146.

Charniak E. (1996). Tree-bank grammars. In *Proceedings of the Thirteenth National Conference on Artificial Intelligence*, pp. 1031–1036.

Charniak E. and Johnson M. (2005). Coarse-to-fine n-best parsing and MaxEnt discriminative reranking. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, Ann Arbor, Michigan. Association for Computational Linguistics, pp. 173–180.

Choi D., Park J. and Choi K.-S. (2012). Korean Treebank transformation for parser training. In *Proceedings of the ACL 2012 Joint Workshop on Statistical Parsing and Semantic Processing of Morphologically Rich Languages*, Jeju, Republic of Korea. Association for Computational Linguistics, pp. 78–88.

Choi J.D. and Palmer M. (2011). Statistical dependency parsing in Korean: From corpus generation to automatic parsing. In *Proceedings of the Second Workshop on Statistical Parsing of Morphologically Rich Languages*, Dublin, Ireland. Association for Computational Linguistics, pp. 1–11.

https://doi.org/10.1017/S1351324920000479 Published online by Cambridge University Press

Choi K.-S., Han Y.S., Han Y.G. and Kwon O.W. (1994). KAIST tree bank project for Korean: Present and future development. In *Proceedings of the International Workshop on Sharable Natural Language Resources*, Nara Institute of Science and Technology. Nara Institute of Science and Technology, pp. 7–14.

Chomsky N. (1981). *Lectures on Government and Binding*. Studies in Generative Grammar. Dordrecht, The Netherlands: Foris Publications.

Chomsky N. (1982). *Some Concepts and Consequences of the Theory of Government and Binding*. *Linguistic Inquiry Monograph*, vol. 6. Cambridge, MA: The MIT Press.

Chomsky N. (1986). *Barriers*. Linguistic Inquiry Monograph, vol. 13. Cambridge, MA: The MIT Press.

Chung T., Post M. and Gildea D. (2010). Factors affecting the accuracy of Korean parsing. In *Proceedings of the NAACL HLT 2010 First Workshop on Statistical Parsing of Morphologically-Rich Languages*, Los Angeles, CA, USA. Association for Computational Linguistics, pp. 49–57.

Coavoux M. and Crabbé B. (2016). Neural greedy constituent parsing with dynamic oracles. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Berlin, Germany. Association for Computational Linguistics, pp. 172–182.

Collins M. (1999). *Head-Driven Statistical Models for Natural Language Parsing*. PhD Thesis, University of Pennsylvania.

Collins M. (2000). Discriminative reranking for natural language parsing. In *Proceedings of the Seventeenth International Conference on Machine Learning*, ICML'00, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc., pp. 175–182.

Collins M. and Koo T. (2015). Discriminative reranking for natural language parsing. *Computational Linguistics* **31**(1), 25–70.

Dakota D. and Kübler S. (2017). Towards replicability in parsing. In *Proceedings of the International Conference Recent Advances in Natural Language Processing, RANLP 2017*, Varna, Bulgaria. INCOMA Ltd., pp. 185–194.

Devlin J., Chang M.-W., Lee K. and Toutanova K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North {A}merican Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, Minneapolis, Minnesota. Association for Computational Linguistics, pp. 4171–4186.

Durrett G. and Klein D. (2015). Neural CRF parsing. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, Beijing, China. Association for Computational Linguistics, pp. 302–312.

Fernández-González D. and Gómez-Rodríguez C. (2018). Dynamic oracles for top-down and in-order shift-reduce constituent parsing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, Brussels, Belgium. Association for Computational Linguistics, pp. 1303–1313.

Fried D., Kitaev N. and Klein D. (2019). Cross-domain generalization of neural constituency parsers. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, Florence, Italy. Association for Computational Linguistics, pp. 323–330.

Fried D. and Klein D. (2018). Policy gradient as a proxy for dynamic oracles in constituency parsing. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, Melbourne, Australia. Association for Computational Linguistics, pp. 469–476.

Fried D., Stern M. and Klein D. (2017). Improving neural parsing by disentangling model combination and reranking effects. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, Vancouver, Canada. Association for Computational Linguistics, pp. 161–166.

Goldberg Y., Sartorio F. and Satta G. (2014). A tabular method for dynamic oracles in transition-based parsing. *Transactions of the Association for Computational Linguistics* **2**(1), 119–130.

Han C.-H., Han N.-R. and Ko E.-S. (2001). Bracketing Guidelines for Penn Korean TreeBank. Technical report, University of Pennsylvania.

Han C.-H., Han N.-R., Ko E.-S., Palmer M. and Yi H. (2002). Penn Korean treebank: Development and evaluation. In *Proceedings of the 16th Pacific Asia Conference on Language, Information and Computation*, Jeju, Korea. Pacific Asia Conference on Language, Information and Computation, pp. 69–78.

Hermjakob U. (2000). Rapid parser development: A machine learning approach for Korean. In *Proceedings of the 1st Meeting of the North American Chapter of the Association for Computational Linguistics*, Seattle, Washington, USA, pp. 118–123.

Hermjakob U. and Mooney R.J. (1997). Learning parse and translation decisions from examples with rich context. In *35th Annual Meeting of the Association for Computational Linguistics and 8th Conference of the European Chapter of the Association for Computational Linguistics*, Madrid, Spain. Association for Computational Linguistics, pp. 482–489.

Johnson M. (1998). PCFG models of linguistic tree representations. *Computational Linguistics* **24**(4), 613–632.

Jung S., Lee C. and Hwang H. (2018). End-to-end Korean part-of-speech tagging using copying mechanism. *ACM Transactions on Asian and Low-Resource Language Information Processing (TALLIP)* **17**(3), 19:1–19:8.

Kitaev N., Cao S. and Klein D. (2019). Multilingual constituency parsing with self-attention and pre-training. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, Florence, Italy. Association for Computational Linguistics, pp. 3499–3505.

https://doi.org/10.1017/S1351324920000479 Published online by Cambridge University Press

Kitaev N. and Klein D. (2018). Constituency parsing with a self-attentive encoder. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Melbourne, Australia. Association for Computational Linguistics, pp. 2675–2685.

Klein D. and Manning C.D. (2001). Parsing with treebank grammars: Empirical bounds, theoretical models, and the structure of the Penn Treebank. In *Proceedings of 39th Annual Meeting of the Association for Computational Linguistics*, Toulouse, France. Association for Computational Linguistics, pp. 338–345.

Klein D. and Manning C.D. (2003). Accurate unlexicalized parsing. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*, Sapporo, Japan. Association for Computational Linguistics, pp. 423–430.

Kulmizev A., Ravishankar V., Abdou M. and Nivre J. (2020). Do neural language models show preferences for syntactic formalisms? In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, Online, 7 2020. Association for Computational Linguistics, pp. 4077–4091.

Kummerfeld J.K., Hall D., Curran J.R. and Klein D. (2012). Parser showdown at the wall street corral: An empirical investigation of error types in parser output. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, Jeju Island, Korea. Association for Computational Linguistics, pp. 1048–1059.

Kummerfeld J.K., Tse D., Curran J.R. and Klein D. (2013). An empirical examination of challenges in Chinese parsing. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, Sofia, Bulgaria. Association for Computational Linguistics, pp. 98–103.

Kuncoro A., Ballesteros M., Kong L., Dyer C., Neubig G. and Smith N.A. (2017). What do recurrent neural network grammars learn about syntax? In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, Valencia, Spain. Association for Computational Linguistics, pp. 1249–1258.

Matsuzaki T., Miyao Y. and Tsujii J. (2005). Probabilistic CFG with latent annotations. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, Ann Arbor, Michigan. Association for Computational Linguistics, pp. 75–82.

Mikolov T., Chen K., Corrado G.S. and Dean J. (2013a). Efficient estimation of word representations in vector space. In *Proceedings of Workshop at the International Conference on Learning Representations (ICLR) 2013*, Scottsdale, Arizona. The International Conference on Learning Representations (ICLR).

Mikolov T., Sutskever I., Chen K., Corrado G.S. and Dean J. (2013b). Distributed representations of words and phrases and their compositionality. In **Burges C.J.C., Bottou L., Welling M., Ghahramani Z. and Weinberger K.Q.** (eds), *Advances in Neural Information Processing Systems*, vol. **26**. Lake Tahoe, Nevada: Curran Associates, Inc., pp. 3111–3119.

Oh J.-Y. and Cha J.-W. (2013). Korean dependency parsing using key Eojoel. *Journal of KIISE: Software and Applications* **40**(10), 600–608.

Park J. (2006). *Extraction automatique d'une grammaire d'arbres adjoints à partir d'un corpus arboré pour le coréen*. PhD Thesis, Université Paris 7 - Denis Diderot, Paris, France.

Park J. (2018). Une note sur l'analyse du constituant pour le français. In *Actes de Traitement Automatique des Langues Naturelles (TANL2018)*, Rennes, France. ATALA, pp. 251–260.

Park J., Dugast L., Hong J.-P., Shin C.-U. and Cha J.-W. (2017). Building a better bitext for structurally different languages through self-training. In *Proceedings of the First Workshop on Curation and Applications of Parallel and Comparable Corpora*, Taipei, Taiwan. Asian Federation of Natural Language Processing, pp. 1–10.

Park J., Hong J.-P. and Cha J.-W. (2016). Korean language resources for everyone. In *Proceedings of the 30th Pacific Asia Conference on Language, Information and Computation: Oral Papers (PACLIC 30)*, Seoul, Korea. Pacific Asia Conference on Language, Information and Computation, pp. 49–58.

Park J., Kawahara D., Kurohashi S. and Choi K.-S. (2013). Towards fully lexicalized dependency parsing for Korean. In *Proceedings of the 13th International Conference on Parsing Technologies (IWPT 2013)*, Nara, Japan. Assocation for Computational Linguistics, pp. 120–126.

Park J. and Tyers F. (2019). A new annotation scheme for the Sejong part-of-speech tagged corpus. In *Proceedings of the 13th Linguistic Annotation Workshop*, Florence, Italy. Association for Computational Linguistics, pp. 195–202.

Pennington J., Socher R. and Manning C.D. (2014). GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Doha, Qatar. Association for Computational Linguistics, pp. 1532–1543.

Petrov S., Barrett L., Thibaux R. and Klein D. (2006). Learning accurate, compact, and interpretable tree annotation. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, Sydney, Australia. Association for Computational Linguistics, pp. 433–440.

Petrov S. and Klein D. (2007). Improved inference for unlexicalized parsing. In *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Proceedings of the Main Conference*, Rochester, New York. Association for Computational Linguistics, pp. 404–411.

Post M. and Gildea D. (2009). Bayesian learning of a tree substitution grammar. In *Proceedings of the ACL-IJCNLP 2009 Conference Short Papers*, Suntec, Singapore. Association for Computational Linguistics, pp. 45–48.

https://doi.org/10.1017/S1351324920000479 Published online by Cambridge University Press

Sarkar A. (2002). *Statistical Parsing Algorithms for Lexicalized Tree Adjoining Grammars*. PhD Thesis, University of Pennsylvania, Philadelphia, Pennsylvania, USA.

Sarkar A. and Han C.-H. (2002). Statistical morphological tagging and parsing of Korean with an LTAG grammar. In *Proceedings of 6th International Workshop on Tree Adjoining Grammars and Related Frameworks (TAG+6)*, Venice, Italy, pp. 48–56.

Seddah D., Kübler S. and Tsarfaty R. (2014). Introducing the SPMRL 2014 shared task on parsing morphologically-rich languages. In *Proceedings of the First Joint Workshop on Statistical Parsing of Morphologically Rich Languages and Syntactic Analysis of Non-Canonical Languages*, Dublin, Ireland. Dublin City University, pp. 103–109.

Seddah D., Tsarfaty R., Kübler S., Candito M., Choi J.D., Farkas R., Foster J., Goenaga I., Gojenola Galletebeitia K., Goldberg Y., Green S., Habash, N., Kuhlmann, M., Maier, W., Nivre, J., Przepiórkowski A., Roth R., Seeker W., Versley Y., Vincze V., Woliński M., Wróblewska A. and de la Clergerie, E.V. (2013). Overview of the SPMRL 2013 shared task: A cross-framework evaluation of parsing morphologically rich languages. In *Proceedings of the Fourth Workshop on Statistical Parsing of Morphologically-Rich Languages*, Seattle, Washington, USA. Association for Computational Linguistics, pp. 146–182.

Socher R., Bauer J., Manning C.D. and Ng A.Y. (2013). Parsing with compositional vector grammars. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Sofia, Bulgaria. Association for Computational Linguistics, pp. 455–465.

Stymne S., Lhoneux M., Smith A. and Nivre J. (2018). Parser training with heterogeneous treebanks. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, Melbourne, Australia. Association for Computational Linguistics, pp. 619–625.

Vinyals O., Kaiser L., Koo T., Petrov S., Sutskever I. and Hinton G.E. (2015). Grammar as a foreign language. In **Cortes C., Lawrence N.D., Lee, D.D., Sugiyama M. and Garnett R.** (eds), *Advances in Neural Information Processing Systems 28*. Curran Associates, Inc., pp. 2773–2781

Watanabe T. and Sumita E. (2015). Transition-based neural constituent parsing. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, Beijing, China. Association for Computational Linguistics, pp. 1169–1179.

Xia F., Han C., Palmer M. and Joshi A. (2000). Comparing lexicalized treebank grammars extracted from Chinese, Korean, and English Corpora. In *Second Chinese Language Processing Workshop*, Hong Kong, China. Association for Computational Linguistics, pp. 52–59.

Cite this article: Kim M and Park J (2022). A note on constituent parsing for Korean. *Natural Language Engineering* **28**, 199–222. https://doi.org/10.1017/S1351324920000479

https://doi.org/10.1017/S1351324920000479 Published online by Cambridge University Press