Development of the Korean Resource Grammar: Towards Grammar Customization

Sanghoun Song
Dept. of Linguistics
Univ. of Washington
sanghoun@uw.edu

Jong-Bok Kim
School of English
Kyung Hee Univ.
jongbok@khu.ac.kr

Francis Bond
Linguistics and Multilingual Studies
Nanyang Technological Univ.
bond@ieee.org

Jaehyung Yang
Computer Engineering
Kangnam Univ.
jhyang@kangnam.ac.kr

Abstract

The Korean Resource Grammar (KRG) is a computational open-source grammar of Korean (Kim and Yang, 2003) that has been constructed within the DELPH-IN consortium since 2003. This paper reports the second phase of the KRG development that moves from a phenomena-based approach to grammar customization using the LinGO Grammar Matrix. This new phase of development not only improves the parsing efficiency but also adds generation capacity, which is necessary for many NLP applications.

1 Introduction

The Korean Resource Grammar (KRG) has been under development since 2003 (Kim and Yang, 2003) with the aim of building an open source grammar of Korean. The grammatical framework for the KRG is Head-driven Phrase Structure Grammar (HPSG: (Pollard and Sag, 1994; Sag et al., 2003)), a non-derivational, constraint-based, and surface-oriented grammatical architecture. The grammar models human languages as systems of constraints on typed feature structures. This enables the extension of grammar in a systematic and efficient way, resulting in linguistically precise and theoretically motivated descriptions of languages.

The initial stage of the KRG (hereafter, KRG1) has covered a large part of the Korean grammar with fine-grained analyses of HPSG. However, this version, focusing on linguistic data with theory-oriented approaches, is unable to yield efficient parsing or generation. The additional limit of the KRG1 is its unattested parsing efficiency with a large scale of naturally occurring data, which is a prerequisite to the practical uses of the developed grammar in the area of MT.

Such weak points have motivated us to move the development of KRG to a data-driven approach from a theory-based one upon which the KRG1 is couched. In particular, this second phase of the KRG (henceforth, KRG2) also starts with two methods: shared grammar libraries (the Grammar Matrix (Bender et al., 2002; Bender et al., 2010)) and data-driven expansion (using the Korean portions of multilingual texts).

Next, we introduce the resources we used (§ 2). this is followed by more detailed motivation for our extensions (§ 3). We then detail how we use the grammar libraries from the Grammar Matrix to enable generation (§ 2) and then expand the coverage based on a corpus study (§ 5).

2 Background

2.1 Open Source NLP with HPSG

The Deep Linguistic Processing with HPSG Initiative (DELPH-IN: `www.delph-in.net`) provides an open-source collection of tools and grammars for deep linguistic processing of human language within the HPSG and MRS (Minimal Recursion Semantics (Copestake et al., 2005)) framework. The resources include software packages, such as the LKB for parsing and generation, PET (Callmeier, 2000) for parsing, and a profiling tool [incr_tsdb()] (Oepen, 2001). There are also several grammars: e.g. ERG; the

Proceedings of the 8th Workshop on Asian Language Resources, pages 144–152,
Beijing, China, 21-22 August 2010. ©2010 Asian Federation for Natural Language Processing

English Resource Grammar (Flickinger, 2000), Jacy; a Japanese Grammar (Siegel and Bender, 2002), the Spanish grammar, and so forth. These along with some pre-compiled versions of preprocessing or experimental tools are packaged in the LOGON distribution.[1] Most resources are under the MIT license, with some parts under other open licenses such as the LGPL.[2] The KRG has been constructed within this open-source infrastructure, and is released under the MIT license[3].

2.2 The Grammar Matrix

The Grammar Matrix (Bender et al., 2002; Bender et al., 2010) offers a well-structured environment for the development of precision-based grammars. This framework plays a role in building a HPSG/MRS-based grammar in a short time, and improving it continuously. The Grammar Matrix covers quite a few linguistic phenomena constructed from a typological view. There is also a starter-kit, the Grammar Matrix customization system which can build the backbone of a computational grammar from a linguistic description.

2.3 A Data-driven Approach

Normally speaking, building up a computational grammar is painstaking work, because it costs too much time and effort to develop a grammar by hand only. An alternative way is a data-driven approach which ensures 'cheap, fast, and easy' development. However, this does not mean that one is better than the other. Each of these two approaches has its own merits. To achieve the best or highest performance of parsing and generation, each needs to complement the other.

3 Directions for Improvement

3.1 Generation for MT

HPSG/MRS-based MT architecture consists of parsing, transfer, and generation, as assumed in Figure 1 (Bond et al., 2005). As noted earlier, the KRG1 with no generation function is limited only to the Source Analysis in Figure 1. In addition, since its first aim was to develop a Korean grammar that reflects its individual properties in detail, the KRG1 lacks compatible semantic representations with other grammars such as ERG and Jacy. The mismatches between the components of the KRG1 and those of other grammars make it difficult to adopt the Korean grammar for an MT system. To take a representative example, the KRG1 treats tense information as a feature type of HEAD, while other grammars incorporate it into the semantics; thus, during the transfer process in Figure 1, some information will be missing. In addition, KRG1 used default inheritance, which makes the grammar more compact, but means it could not used with the faster PET parser. We will discuss this issue in more detail in Section 4.1.

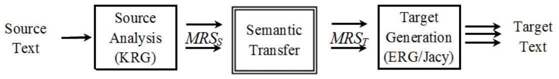

Figure 1: HPSG/MRS-based MT Architecture

Another main issue in the KRG1 is that some of the defined types and rules in the grammar are inefficient in generation. Because the declared types and rules are defined with theoretical motivations, the run-time for generating any parsing units within the system takes more than expected and further causes memory overflow errors to crop up almost invariably, even though the input is quite simple. This problem is partially due to the complex morphological inflection system in the KRG1. Section 4.2 discusses how KRG2, solves this problem.

Third it is better "to be robust for parsing and strict for generation" (Bond et al., 2008). That means robust rules will apply in parsing, though the input sentence does not sound perfect, but not in generation. For example, the sentence (1b), the colloquial form of the formal, standard sentence (1a), is used more frequently in colloquial context:

(1) a. ney-ka cham yeppu-ney.
you-NOM really pretty-DECL
'You are really pretty.'
b. ni-ka cham ippu-ney

The grammar needs to parse both (1a) and

[1] wiki.delph-in.net/moin/LogonTop

[2] www.opensource.org/licenses/

[3] It allows people "...without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies" so long as "The above copyright notice and this permission notice shall be included ..."

(1b) and needs to yield the same MRS because both sentences convey the same truth-conditional meaning. However, the KRG1 handles only the legitimate sentence (1a), excluding (1b). The KRG1 is thus not sophisticated enough to distinguish these two stylistic different sentences. Therefore we need to develop the generation procedures that can choose a proper sentence style. Section 4.3 proposes the 'STYLE' feature structure as the choice device.

3.2 Exploiting Corpora

One of the main motivations for our grammar improvement is to achieve more balance between linguistic motivation and practical purpose. We have first evaluated the coverage and performance of the KRG1 using a large size of data to track down the KRG1's problems that may cause parsing inefficiencies and generating clog. In other words, referring to the experimental results, we patterned the problematic parts in the current version. According to the error pattern, on the one hand, we expanded lexicon from occurring texts in our generalization. On the other hand, we fixed the previous rules and sometimes introduced new rules with reference to the occurrence in texts.

3.3 How to Improve

In developing the KRG, we have employed two strategies for improvement; (i) shared grammar libraries and (ii) exploiting large text corpora.

We share grammar libraries with the Grammar Matrix in the grammar (Bender et al., 2002) as the foundation of KRG2. The Grammar Matrix provides types and constraints that assist the grammar in producing well-formed MRS representations. The Grammar Matrix customization system provides with a linguistically-motivated broad coverage grammar for Korean as well as the basis for multilingual grammar engineering. In addition, we exploit naturally occurring texts as the generalization corpus. We chose as our corpora Korean texts that have translations available in English or Japanese, because they can be the baseline of multilingual MT. Since the data-driven approach is influenced by data type, multilingual texts help us make the grammar more suitable for MT in the long term. In developing the grammar in the next phrase, we assumed the following principles:

(2) a. The Grammar Matrix will apply when a judgment about structure (e.g. semantic representation) is needed.

b. The KRG will apply when a judgment about Korean is needed.

c. The resulting grammar has to run on both PET and LKB without any problems.

d. Parsing needs to be accomplished as robustly as possible, and generation needs to be done as strictly as possible.

4 Generation

It is hard to alter the structure of the KRG1 from top to bottom in a relatively short time, mainly because the difficulties arise from converting each grammar module (optimized only for parsing) into something applicable to generation, and further from making the grammar run separately for parsing and generation.

Therefore, we first rebuilt the basic schema of the KRG1 on the Grammar Matrix customization system, and then imported each grammar module from KRG1 to the matrix-based frame (§4.1). In addition, we reformed the inflectional hierarchy assumed in the KRG1, so that the grammar does not impede generation any longer (§ 4.2). Finally, we introduced the STYLE feature structure for sentence choice in accordance with our principles (2c-d) (§4.3).

4.1 Modifying the Modular Structure

The root folder `krg` contains the basic type definition language files (`*.tdl`. In the KRG2, we subdivided the `types.tdl` into: `matrix.tdl` file which corresponds to general principles; `korean.tdl` with language particular rules; `types-lex.tdl` for lexical types and `types-ph.tdl` for phrasal types. In addition, we reorganized the KRG1's `lexicons.tdl` file into the `lex` folder consisting of several sub-files in accordance with the POS values (e.g.; `lex-v.tdl` for verbs).

The next step is to revise grammar modules in order to use the Grammar Matrix to a full extent. In this process, when inconsistencies arise between KRG1 and KRG2, we followed (2a-b).

We further transplanted each previous module into the KRG2, while checking the attested test items used in the KRG1. The test items, consisting of 6,180 grammatical sentences, 118 ungrammatical sentences, were divided into subgroups according to the related phenomena (e.g. light verb constructions).

4.2 Simplifying the Inflectional Hierarchy

Korean has rigid ordering restrictions in the morphological paradigm for verbs, as shown in (3).

(3) a. V-base + HON + TNS + MOOD + COMP
b. ka-si-ess-ta-ko 'go-HON-PST-DC-COMP'

KRG1 dealt with this ordering of suffixes by using a type hierarchy that represents a chain of inflectional slots (Figure 2: Kim and Yang (2004)).

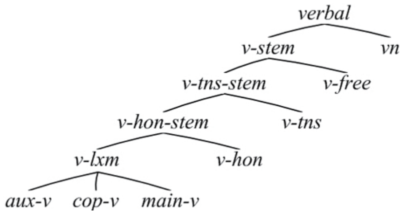

Figure 2: Korean Verbal Hierarchy

This hierarchy has its own merits, but it is not so effective for generating sentences. This is because the hierarchy requires a large number of calculations in the generation process. Figure 3 and Table 1 explains the difference in computational complexity according to each structure.In Figure 3, (a) is similar to Figure 2, while (b) is on the traditional template approach. Let us compare each complexity to get the target node D. For convenience' sake, let us assume that each node has ten constraints to be satisfied. In (a), since there are three parents nodes (i.e. A, B, and C) on top of D, D cannot be generated until A, B, and C are checked previously. Hence, it costs at least 10,000 (10[A] ×10[B] ×10[C] ×10[D]) calculations. In contrast, in (b), only 100 (10[A] ×10[D]) calculations is enough to generate node D. That means, the deeper the hierarchy is, the more the complexity increases. Table 1 shows (a) requires more than 52 times as much complexity as (b), though they have the same number of nodes.

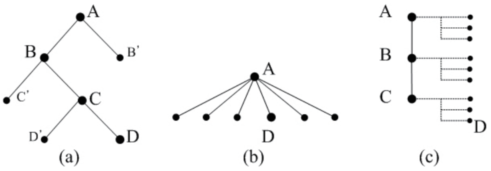

Figure 3: Calculating Complexity

Table 1: Complexity of (a) and (b)

(a)			(b)
B'	10[A]×10[B']	100	10[A]×10[B']
C'	10[A]×10[B]×10[C']	1,000	10[A]×10[C']
D'	10[A]×10[B]×10[C]×10[D']	10,000	10[A]×10[D']
D	10[A]×10[B]×10[C]×10[D]	10,000	10[A]×10[D]
Σ	21,100		400

When generation is processed by LKB, all potential inflectional nodes are made before syntactic configurations according to the given MRS. Thus, if the hierarchy becomes deeper and contains more nodes, complexity of (a)-styled hierarchy grows almost by geometric progression. This makes generation virtually impossible, causing memory overflow errors to the generation within the KRG1.

A fully flat structure (b) is not always superior to (a). First of all, the flat approach ignores the fact that Korean is an agglutinative language. Korean morphological paradigm can yield a wide variety of forms; therefore, to enumerate all potential forms is not only undesirable but also even impossible.

The KRG2 thus follows a hybrid approach (c) that takes each advantage of (a) and (b). (c) is more flattened than (a), which lessens computational complexity. On the other hand, in (c), the depth of the inflectional hierarchy is fixed as two, and the skeleton looks like a unary form, though each major node (marked as a bigger circle) has its own subtypes (marked as dotted lines). Even though the depth has been diminished, the hierarchy is not a perfectly flat structure; therefore, it can partially represent the austere suffix ordering in Korean. The hierarchy (c), hereby, curtails the cost of generation.

In this context, we sought to use the minimum number of possible inflectional slots for Korean. We need at least three: root + semantic slot(s) + syntactic slot(s). That is, a series of suffixes

Table 2: Complexity of (a-c)

	Depth	Complexity
(a)	n ≥ 3	≥ 10,000
(b)	n = 1	100
(c)	n = 2	10,000

that denote semantic information attaches to the second slot, and a series of suffixes, likewise, attaches to the third slot. Since semantic suffixes are, almost invariably, followed by syntactic ones in Korean, this ordering is convincing, granting that it does not fully represent that there is also an ordering among semantic forms or syntactic ones. (4) is an example from hierarchy (c). There are three slots; root *ka* 'go', semantic suffixes *si-ess*, and syntactic ones *ta-ko*.

(4) a. V-base + (HON+TNS) + (MOOD+COMP)

b. ka-si+ess-ta+ko 'go-HON+PST-DC+COMP'

Assuming there are ten constraints on each node, the complexity to generate D in (c) is just 10,000. The measure, of course, is bigger than that of (b), but the number never increases any more. That means, all forms at the same depth have equal complexity, and it is fully predictable. Table 2 compares the complexity from (a) to (c). By converting (a) to (c), we made it possible to generate with KRG2.

4.3 Choosing a Sentence Style

The choice between formal or informal (colloquial) sentence styles depends on context. A robust parser should cover both styles, but we generally want a consistent style when generating.

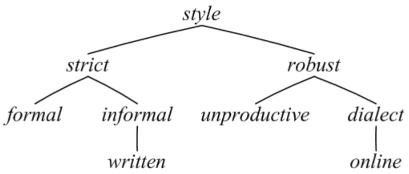

Figure 4: Type Hierarchy of STYLE

In such a case, the grammar resorts to STYLE to filter out the infelicitous results. The type hierarchy is sketched out in Figure 4. *strict* is near to school grammar (e.g. *written* is a style of newspapers). On the other hand, some variant forms that stem from the corresponding canonical forms falls under *robust* in Figure 4. For instance, if the text domain for generation is newspaper, we can select only *written* as our sentence choice, which excludes other styled sentences from our result.

Let us see (1a-b) again. *ni* 'you' in (1b) is a dialect form of *ney*, but it has been used more productively than its canonical form in daily speech. In that case, we can specify STYLE of *ni* as *dialect* as given below. In contrast, the neutral form *ney* has an unspecified STYLE feature:

```
ni := n-pn-2nd-non-pl &
[ STEM < ``ni'' >, STYLE dialect ].
ney := n-pn-2nd-non-pl &
[ STEM < ``ney'' > ].
```

Likewise, since the predicate in (1b) *ippu* 'pretty' stems from *yeppu* in (1a), they share the predicate name '*_yeppu_a_1_rel*' (i.e. the RMRS standard for predicate names such as '_lemma_pos_sense_rel'), but differ in each STYLE feature. That means (1a-b) share the same MRS structure (given below). KRG hereby can parse (1b) into the same MRS as (1a) and generate (1a) from it.

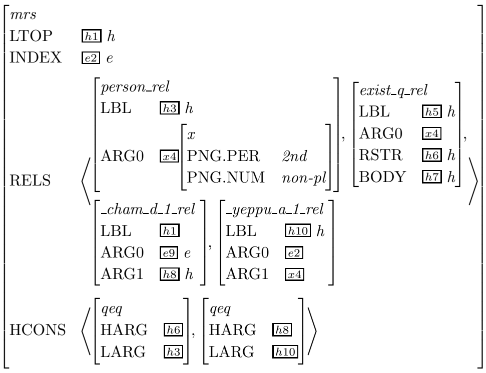

Figure 5: MRS of (1a-b)

These kinds of stylistic differences can take place at the level of (i) lexicon, (ii) morphological combination, and (iii) syntactic configuration. The KRG2 revised each rule with reference to its style type; therefore, we obtained totally 96 robust rules. As a welcoming result, we could manipulate our generation, which was successful respect to (2c-d). Let us call the version reconstructed so far '**base**'.

5 Exploiting Corpora

5.1 Resources

This study uses two multilingual corpora; one is the *Sejong Bilingual Corpora*: SBC (Kim and Cho, 2001), and the other is the *Basic Travel Expression Corpus*: BTEC (Kikui et al., 2003). We exploited the Korean parts in each corpus, taking them as our generalization corpus. Table 3 represents the configuration of two resources (KoEn: Korean-English, KoJa: Korean-Japanese):

Table 3: Generalization Corpora

	SBC	**BTEC**
Type	Bilingual	Multilingual
Domain	Balanced Corpus	Tourism
Words	KoEn : 243,788 KoJa : 276.152	914,199
T/T ratio	KoEn : 27.63 KoJa : 20.28	92.7
Avr length	KoEn : 16.30 KoJa : 23.30	8.46

We also make use of nine test suites sorted by three types (Each test suite includes 500 sentences). As the first type, we used three test sets covering overall sentence structures in Korean; Korean Phrase Structure Grammar (**kpsg**; Kim (2004)), Information-based Korean Grammar (**ibkg**; Chang (1995)), and the SERI test set (**seri**; Sung and Jang (1997)).

Second, we randomly extracted sentences from each corpus, separately from our generalization corpus; two suites were taken from the Korean-English and Korean-Japanese pair in SBC (**sj-ke** and **sj-kj**, respectively). The other two suites are from the BTEC-KTEXT (**b-k**), and the BTEC-CSTAR (**b-c**); the former consists of relatively plain sentences, while the latter is composed of spoken ones.

Third, we obtained two test suites from sample sentences in two dictionaries; Korean-English (**dic-ke**), and Korean-Japanese (**dic-kj**). These suites assume to have at least two advantages with respect to our evaluation; (i) the sentence length is longer than that of BTEC as well as shorter than that of SBC, (ii) the sample sentences on dictionaries are normally made up of useful expressions for translation.

5.2 Methods

We tried to do experiments and improve the KRG, following the three steps repeatedly: (i) evaluating, (ii) identifying, and (iii) exploiting. In each of the first step, we tried to parse the nine test suites and generate sentences with the MRS structures obtained from the parsing results, and measured their coverage and performance. Here, 'coverage' means how many sentences can be parsed or generated, and 'performance' represents how many seconds it takes on average. In the second step, we identified the most serious problems. In the third step, we sought to exploit our generalization corpora in order to remedy the drawbacks. After that, we repeated the procedures until we obtain the desired results.

5.3 Experiments

So far, we have got two versions; **KRG1** and **base**. Our further experiments consist of four phases; **lex**, **MRS**, **irules**, and **KRG2**.

Expanding the lexicon: To begin with, in order to broaden our coverage, we expanded our lexical entries with reference to our generalization corpus and previous literature. Verbal items are taken from Song (2007) and Song and Choe (2008), which classify argument structures of Korean verbal lexicon into subtypes within the HPSG framework in a semi-automatic way. The reason why we do not use our corpus here is that verbal lexicon commonly requires subcategorization frames, but we cannot induce them so easily only using corpora. For other word classes, we extracted lexical items from the POS tagged SBC and BTEC corpora. Table 4 explains how many items we extracted from our generalization corpus. Let us call this version '**lex**'.

Table 4: Expansion of Lexical Items

verbal nouns	4,474
verbs and adjectives	1,216
common nouns	11,752
proper nouns	7,799
adverbs	1,757
numeral words	1,172

MRS: Generation in LKB, as shown in Figure 1, deploys MRS as the input, which means our generation performance hinges on the well-formedness of MRS. In other words, if our MRS is broken somewhere or constructed inefficiently, generation results is directly affected. For instance, if the semantic representation does not scope, we will not generate correctly. We were able to identify such sentences by parsing the corpora, storing the semantic representations and then using the semantic well formedness checkers in the LKB. We identified all rules and lexical items that produced ill-formed MRSs using a small script and fixed them by hand. This had an immediate and positive effect on coverage as well as performance in generation. We refer to these changes as '**MRS**'.

Different inflectional forms for sentence styles: Texts in our daily life are actually composed of various styles. For example, spoken forms are normally more or less different from written ones. The difference between them in Korean is so big that the current version of KRG can hardly parse spoken forms. Besides, Korean has lots of compound nouns and derived words. Therefore, we included these forms into our inflectional rules and expanded lexical entries again (3,860 compound nouns, 2,791 derived words). This greatly increased parsing coverage. We call this version '**irules**'.

Grammaticalized and Lexicalized Forms: There are still remaining problems, because our test suites contain some considerable forms. First, Korean has quite a few grammaticalized forms; for instance, *kupwun* is composed of a definite determiner *ku* and a classifier for human *pwun* "the person", but it functions like a single word (i.e. a third singular personal pronoun). In a similar vein, there are not a few lexicalized forms as well; for example, a verbal lexeme *kkamek-* is composed of *kka-* "peel" and *mek-* "eat", but it conveys a sense of "forget", rather than "peel and eat". In addition, we also need to cover idiomatic expressions (e.g. "thanks") for robust parsing. Exploiting our corpus, we added 1,720 grammaticalized or lexicalized forms and 352 idioms. Now, we call this '**KRG2**'.

Table 5 compares KRG2 with KRG1, and Figure 6 shows how many lexical items we have covered so far.

Table 5: Comparison between KRG1 and KRG2

	KRG1	KRG2
# of default types	121	159
# of lexical types	289	593
# of phrasal types	58	106
# of inflectional rules	86	244
# of syntactic rules	36	96
# of lexicon	2,297	39,190

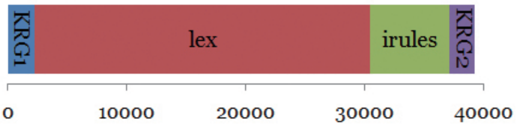

Figure 6: Size of Lexicon

5.4 Evaluation

Table 6 shows the evaluation measure of this study. 'p' and 'g' stand for 'parsing' and 'generation', respectively. '+' represents the difference compared to KRG1. Since KRG1 does not generate, there is no 'g+'.

Table 6: Evaluation

	coverage (%)				ambiguity	
	p	p+	g	s	p	g
kpsg	77.0	-5.5	55.2	42.5	174.9	144.4
ibkg	61.2	41.8	68.3	41.8	990.5	303.5
seri	71.3	-0.8	65.7	46.8	289.1	128.4
b-k	43.0	32.6	62.8	27.0	1769.4	90.0
b-c	52.2	45.8	59.4	31.0	1175.8	160.6
sj-ke	35.4	31.2	58.2	20.6	358.3	170.3
sj-kj	23.0	19.6	52.2	12.0	585.9	294.9
dic-ke	40.4	31.0	42.6	17.2	1392.7	215.9
dic-kj	34.8	25.2	67.8	23.6	789.3	277.9
avr	**48.7**	**24.5**	**59.1**	**28.8**	**836.2**	**198.4**

On average, the parsing coverage increases **24.5%**. The reason why there are negative values in 'p+' of **kpsg** and **seri** is that we discarded some modules that run counter efficient processing (e.g., the grammar module for handling floating quantifiers sometimes produces too many ambiguities.). Since KRG1 has been constructed largely around the test sets, we expected it to perform well here. If we measure the parsing coverage again, after excluding the results of **kpsg** and **seri**, it accounts for **32.5%**.[4] The generation coverage of KRG2 accounts for almost **60%** per parsed sentence on average. Note that KRG1 could not parse at all. 's' (short for 'success') means the portion of both parsed and generated sentences (i.e. 'p'×'g'), which accounts

[4] The running times, meanwhile, becomes slower as we would expect for a grammar with greater coverage. However, we can make up for it using the PET parser, as shown in Figure 9.

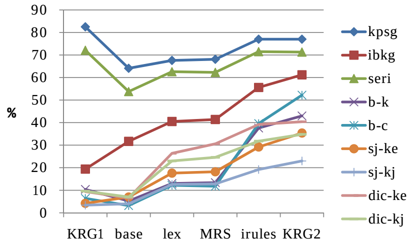

Figure 7: Parsing Coverage (%)

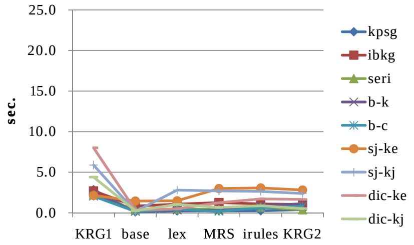

Figure 9: Parsing Performance (s)

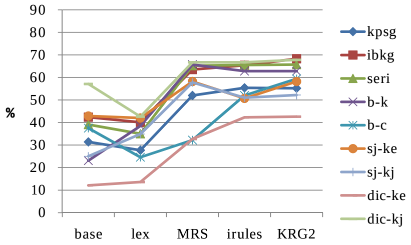

Figure 8: Generation Coverage (%)

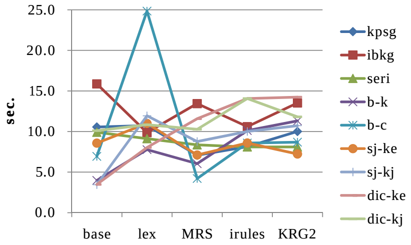

Figure 10: Generation Performance (s)

for about **29%**. Ambiguity means '# of parses/# of sentences' for parsing and '# of realizations/# of MRSes' for generation. The numbers look rather big, which should be narrowed down in our future study.

In addition, we can find out in Table 6 that there is a coverage ordering with respect to the type of test suites; 'test sets > BTEC > dic > SBC'. It is influenced by three factors; (i) lexical variety, (ii) sentence length, and (iii) text domain. This difference implies that it is highly necessary to use variegated texts in order to improve grammar in a comprehensive way.

Figure 7 to 10 represent how much each experiment in §5.3 contributes to improvement. First, let us see Figure 7 and 8. As we anticipated, **lex** and **irules** contribute greatly to the growth of parsing coverage. In particular, the line of **b-c** in Figure 8, which mostly consists of spoken forms, rises rapidly in **irules** and **KRG2**. That implies Korean parsing largely depends on richness of lexical rules. On the other hand, as we also expected, **MRS** makes a great contribution to generation coverage (Figure 8). In **MRS**, the growth accounts for **22%** on average. That implies testing with large corpora must take precedence in order for coverage to grow.

Figure 9 and 10 shows performance in parsing and generation, respectively. Comparing to **KRG1**, our Matrix-based grammars (from **base** to **KRG2**) yields fairly good performance. It is mainly because we deployed the PET parser that runs fast, whereas KRG1 runs only on LKB. Figure 10, on the other hand, shows that the revision of MRS also does much to enhance generation performance, in common with coverage mentioned before. It decreases the running times by about **3.1** seconds on average.

6 Conclusion

The newly developed KRG2 has been successfully included in the LOGON repository since July, 2009; thus, it is readily available. In future research, we plan to apply the grammar in an MT system (for which we already have a prototype). In order to achieve this goal, we need to construct multilingual treebanks; Korean (KRG), English (ERG), and Japanese (Jacy).

Acknowledgments

We thank Emily M. Bender, Dan Flickinger, Jae-Woong Choe, Kiyotaka Uchimoto, Eric Nichols, Darren Scott Appling, and Stephan Oepen for comments and suggestions at various stages. Parts of this research was conducted while the first and third authors were at the National Institute for Information and Communications Technologies (NICT), Japan; we thank NICT for their support. Our thanks also go to three anonymous reviewers for helpful feedback.

References

Bender, Emily M., Dan Flickinger, and Stephan Oepen. 2002. The Grammar Matrix: An Open-Source Starter-Kit for the Rapid Development of Cross-Linguistically Consistent Broad-Coverage Precision Grammars. In *Procedings of the Workshop on Grammar Engineering and Evaluation at the 19th International Conference on Computational Linguistics*.

Bender, Emily M., Scott Drellishak, Antske Fokkens, Michael Wayne Goodman, Daniel P. Mills, Laurie Poulson, and Safiyyah Saleem. 2010. Grammar Prototyping and Testing with the LinGO Grammar Matrix Customization System. In *Proceedings of ACL 2010 Software Demonstrations*.

Bond, Francis, Stephan Oepen, Melanie Siegel, Ann Copestake, and Dan Flickinger. 2005. Open Source Machine Translation with DELPH-IN. In *Proceedings of Open-Source Machine Translation: Workshop at MT Summit X*.

Bond, Francis, Eric Nichols, Darren Scott Appling, and Michael Paul. 2008. Improving Statistical Machine Translation by Paraphrasing the Training Data. In *Proceedings of the 5th International Workshop on Spoken Languaeg Translation*.

Callmeier, Ulrich. 2000. PET–a Platform for Experimentation with Efficient HPSG Processing Techniques. *Natural Language Engineering*, 6(1):99–107.

Chang, Suk-Jin. 1995. *Information-based Korean Grammar*. Hanshin Publishing, Seoul.

Copestake, Ann, Dan Flickinger, Carl Pollard, and Ivan A. Sag. 2005. Minimal Recursion Semantics: An Introduction. *Research on Language and Computation*, 3(4):281–332.

Flickinger, Dan. 2000. On Building a More Efficient Grammar by Exploiting Types. *Natural Language Engineering*, 6 (1) (Special Issue on Efficient Processing with HPSG):15 – 28.

Kikui, Genichiro, Eiichiro Sumita, Toshiyuki Takezawa, and Seiichi Yamamoto. 2003. Creating corpora for speech-to-speech translation. In *Proc. of the EUROSPEECH03*, pages 381–384, Geneve, Switzerland.

Kim, Se-jung and Nam-ho Cho. 2001. The progress and prospect of the 21st century Sejong project. In *ICCPOL-2001*, pages 9–12, Seoul.

Kim, Jong-Bok and Jaehyung Yang. 2003. Korean Phrase Structure Grammar and Its Implementations into the LKB System. In *Proceedings of the 17th Pacific Asia Conference on Language, Information and Computation*.

Kim, Jong-Bok and Jaehyung Yang. 2004. Projections from Morphology to Syntax in the Korean Resource Grammar: Implementing Typed Feature Structures. In *Lecture Notes in Computer Science*, volume 2945, pages 13–24. Springer-Verlag.

Kim, Jong-Bok. 2004. *Korean Phrase Structure Grammar*. Hankwuk Publishing, Seoul.

Oepen, Stephan. 2001. [incr tsdb()] — competence and performance laboratory. User manual. Technical report, Computational Linguistics, Saarland University.

Pollard, Carl and Ivan A. Sag. 1994. *Head-Driven Phrase Structure Grammar*. The University of Chicago Press, Chicago, IL.

Sag, Ivan A., Thomas Wasow, , and Emily M. Bender. 2003. *Syntactic Theory: A Formal Introduction*. CSLI Publications, Stanford, CA.

Siegel, Melanie and Emily M. Bender. 2002. Efficient Deep Processing of Japanese. In *Proceedings of the 3rd Workshop on Asian Language Resources and International Standardization*.

Song, Sanghoun and Jae-Woong Choe. 2008. Automatic Construction of Korean Verbal Type Hierarchy using Treebank. In *Proceedings of HPSG2008*.

Song, Sanghoun. 2007. A Constraint-based Analysis of Passive Constructions in Korean. Master's thesis, Korea University, Department of Linguistics.

Sung, Won-Kyung and Myung-Gil Jang. 1997. SERI Test Suites '95. In *Proceedings of the Conference on Hanguel and Korean Language Information Processing*.