Master's Thesis of Linguistics

Automatic Generation of Morpheme-Level Reordering Rules for Korean to English Machine Translation

한국어-영어 기계 번역을 위한
형태소 수준 재배열 규칙 자동 생성

February 2017

Graduate School of Seoul National University
Department of Linguistics

Breanna Castellani

Automatic Generation of Morpheme-Level Reordering Rules for Korean to English Machine Translation

Hyopil Shin

Submission of a master's thesis in Linguistics

February 2017

Graduate School of Humanities
Seoul National University
Linguistics Major

Breanna Castellani

Confirming the master's thesis written by
Breanna Castellani
February 2017

Chair _____(Seal)

Vice Chair _____(Seal)

Examiner _____(Seal)

Abstract

Word order is one of the main challenges that machine translation systems must overcome when dealing with any linguistically divergent language pair, such as Korean and English. Statistical machine translation (SMT) models are often insufficient at long distance reordering due the distortion penalties they impose. Rule-based systems, on the other hand, are often costly, in both time and money, to build and maintain.

The present research proposes a new hybrid approach for Korean to English machine translation. While previous approaches have focused on the word, our approach considers the morpheme as the basic unit of translation for this language pair. We begin by developing a classification model to disambiguate Korean functional morphemes based on alignment pairs and context feature data. Then, according to our automatically generated rules, we apply this model in a preprocessing step to reorder the morphemes to better match English sentence structure.

After retraining our statistical translation system, Moses, results indicate an improvement in overall translation quality. When the SMT system's internal lexicalized reordering is restricted, we note an increase in the BLEU score of 3.5% over the SMT-only baseline. In the case where we do not limit decoding-time reordering, an even greater BLEU score increase of 4.42% is observed. We also find evidence to suggest that our changes enable Moses to execute additional reordering operations at decoding time that it was previously unable to perform.

Keyword : automatic rule generation, Korean-English MT, hybrid machine translation, rule-based preprocessing, morpheme reordering

Student Number : 2014-25108

Table of Contents

Chapter 1. Introduction

Machine translation (MT) has been an acknowledged area of study in the field of Natural Language Processing (NLP) since the 1950s (Bar-Hillel 1951). It was not until the 1980s, however, when computer processing became exponentially faster and less prohibitively expensive that the research field really saw significant growth (Hutchins 2007). And now, in the 21st century, as the world is becoming increasingly connected due to globalization, the need for efficient and accurate translation of myriad human languages, too, is growing.

MT, conceptually, is a system that receives input in a source language, performs translation algorithms such as decoding, and produces output in a target language. The two main types of translation systems used in recent research are Statistical Machine Translation (SMT) and Rule-based Machine Translation (RBMT) systems (Hutchins 2007). As their names suggest, SMT is a probabilistic approach which employs a statistical model to convert a source sentence to a target sentence, while RBMT utilizes grammar rules, often manually-generated. There are advantages and disadvantages to each, and hybrid approaches, where one method is supplemented with the other in either a pre- or post-processing step, have become a common way to leverage the advantages both, while minimizing the disadvantages of only realizing one by itself. These three approaches will be described in more detail in Chapter 2.

Regardless of the type of machine translation employed, there is one major challenge that must be overcome: word order. All languages have a basic word order, which is typically described using the locations of the subject (S), object (O), and verb (V) in the sentence. English, Russian, and Mandarin Chinese,

for example, are SVO languages, where the subject of the sentence comes before the verb, and the object comes after (Dryer 1991). On the other hand, in languages such as Korean and Japanese, the verb is preceded by both the subject and the object, in a SOV order. Besides these higher level differences, other variations exist, such *head directionality*. This term describes the relative position of a phrasal head to its complement, e.g. head-final or head-initial. Another area of divergence is in regard to *adposition* location, that is, whether a language's temporal and spatial expressions fall before, after, or around their complement (pre-, post-, or circum-positions, respectively) (Dryer 1992).

In order to maximize translation quality, units of translation in the source language must be properly mapped to, or aligned with, those in the target language. Word alignment accuracy, however, is dependent on the relative location of these translation units (Vogel 2003). Thus, as there is potential for word order variation even between languages generally considered more typologically similar, such as English and German, word reordering is an important processing step in machine translation. It is especially necessary when the language pair is linguistically divergent, as is the case with the pair examined in the present study: Korean and English. Korean is considered an SOV language with relatively free word order, but English is SVO with a relatively strict word order. Korean is also head-final and postpositional, while English is head-initial and prepositional. Therefore, correctly compensating for these differences by performing some amount of reordering increases the potential for improved output (Vogel 2003).

There exist two types of word reordering: local, or phrase-internal, and global, or phrase-external (Rottman & Vogel 2007). The former deals with short-distance movement due to change in adposition location or head-directionality,

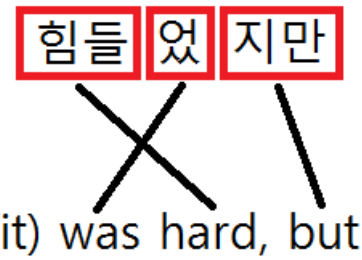

Figure 1. Korean to English word alignment

while the latter handles long-distance movements, such as shifting from SOV to SVO or vice versa. Phrase-based SMT research, described in detail later, has looked at phrase-internal word reordering models using phrase pairs. Phrase-based SMT systems, however, often prove insufficient when it comes to phrase-external reordering, as the distortion models they implement often inflict large penalties for longer movements. Here, for long-distance reordering, rule-based translation systems, which take advantage of information inherent to each language's grammar, prevail (Rottman & Vogel 2007).

Still, neither of these alone adequately process the intricate relationship between Korean and English in MT. As mentioned previously, there are both phrase-internal and phrase-external differences to be captured and considered when translating this particular language pair. Korean itself also presents a somewhat unique challenge. As an agglutinative language, the *morpheme*, not the word, is the basic unit of meaning. In Korean, morphemes combine to create a 'word', which can correspond to one or more 'words' in English (Sohn 2001). Figure 1 shows how a single Korean word can be translated as three separate words in English. In particular, functional morphemes which encode information such as grammatical function, are also frequently highly context-dependent. That is,

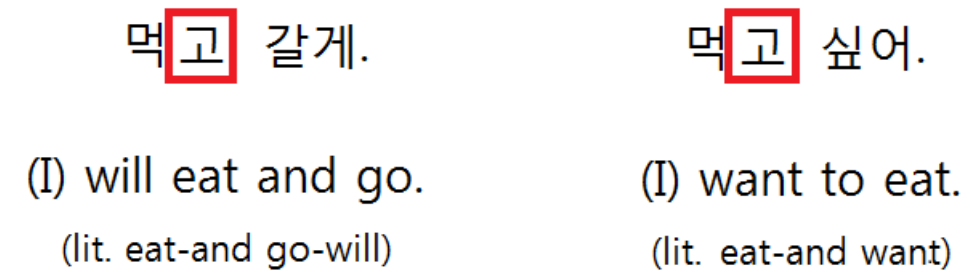

Figure 2. Two parallel sentences with the conjunctive ending -고 'and'

depending on other factors, the same Korean morpheme can be translated differently, and, more importantly, these translations can be found in different locations in the English target sentence. Figure 2 illustrates this problem with two declarative sentences containing the conjunctive ending -고 'and'. In the former, 먹고 갈게, the morpheme retains its meaning of 'and' in the English translation, and still appears inter-verbally. In the latter, however, when it appears before 싶다 'to want', the meaning of 'and' becomes less apparent, and it does not appear in the English translation at all. Evidently, considering only the morpheme itself without other factors like context leads to ambiguity in translation.

Detailed in Chapter 2, most previous research has either focused on the word as the basic meaningful unit for translation (Herrmann et al. 2013, Genzel 2010, among others), or has attempted translating the *less* morphologically complex language into the *more* morphologically complex language (e.g. English to Turkish) (El-Kahlout & Oflazer 2010). Thus, this research proposes a new hybrid approach, which first aims to create a classification model using the original, unmodified bilingual corpus provided by Samsung, to disambiguate Korean functional morphemes based on aggregate context feature data. Then, employing

this fine-grained model in a preprocessing step, we reorder the morphemes according to automatically generated rules. This reordered input is then used to train the SMT system Moses, to produce translations with demonstrable quantitative improvements, assessed using the MT standard evaluation metric, the BLEU score.

To the best of our knowledge, the present research constitutes the first attempt at morpheme-level reordering in Korean to English MT. It is also unique in that the reordering rule process, from start to finish, is done automatically. It is the hope that this automated process can be expanded upon in further research to include other types of morphemes, or even other language pairs.

The structure of this paper is as follows. Chapter 2 provides a summary of the previous research which, taken together, motivated the present research. Chapter 3 briefly describes the corpus data and the SMT system used. Chapter 4 details the reordering process featured in this study, from the development of the morpheme classification model, to the automatic generation of the reordering rules, to the application of these rules. Chapter 5 presents the results and provides a comparison to recent Korean-English research. Chapter 6 is the conclusion.

Chapter 2. Literature Review

In this chapter, we discuss previous research in the general area of machine translation, and, in particular, in the scope of word- and morpheme-level reordering, as well as Korean to English machine translation. We also highlight the areas in which these approaches can be built upon and improved for Korean to English machine translation in the current research.

2.1 Machine Translation

In the introduction, we defined statistical machine translation (SMT) as a probabilistic approach which employs a statistical model to convert a source sentence to a target sentence. Bayes' rule is used to calculate the sentence-level translation probability from source *f* to target *e*, such that:

$$\text{argmax}_e p(e|f) = \text{argmax}_e p(f|e)p(e) \quad (1)$$

Here, the best sentence translation is the candidate translation with the highest conditional probability. However, due to the limitless number of potential sentence configurations, calculating probabilities in this manner, without constraint, is computationally impractical.

Beginning with Berger et al. (1996) and Wu (1996), restrictions were placed on SMT decoders with regard to word alignments. Wu (1996) placed contiguity restraints on the alignments: two words could either maintain the same order or the order could be inverted. Meanwhile, Berger et al. (1996) limited

reordering to at most *n* words at a time. These approaches made the assumption that words that are near to each other in the source sentence should also be nearby in the target sentence. These *n*-gram-based methods gave way to phrase-based research.

Phrase-based SMT breaks the sentence down into smaller, more manageable fragments: *n*-length phrases, consisting of *n* words in sequence. Thus, the translation probabilities can be calculated at the phrase-level, thereby reducing the near-infinite number of sentence configurations found when translating at the word level. Research in this area began with Koehn et al. (2003), who attempted to determine the best phrase-based translation model. Their findings have lead to several staple features in present-day SMT decoders. First, it was found that restricting the potential phrase pairs to only syntactic phrases eliminated too many alignment pairs. The researchers introduced a weight factor which gave preference to linguistic phrases, but did not completely remove non-syntactic phrases from the pool of source-target phrase alignment pairs. Researchers also introduced a distortion cost, which penalizes longer movements linearly by reordering distance. Unfortunately, operating with higher distortion limits, which are needed to make necessary and appropriate reorderings in divergent languages like Korean and English, causes a decrease in translation quality. Thus, SMT approaches are more appropriate for short-distance movements, ideally movements of entire linguistic phrases.

On the other hand, rule-based machine translation (RBMT) is better equipped to handle long distance movements. In RBMT, the translation rules are based on a detailed analysis of the source language. These rules can accurately reflect the language's syntax, morphology, and semantics, and are therefore able to

deal with more complicated reordering conditions, such as phrase-external movement. However, due to their detailed nature, these rules are often hand-written (Collins et al. 2005, Wang et al. 2007, Lee et al. 2010, among others), which takes time and manual effort by knowledgeable linguists. Additionally, because the rules are often domain-specific, reusability is low and maintenance costs are often expensive.

More commonly, however, a RBMT approach is not used solely by itself, but rather, it is combined with SMT in a hybrid approach. Typically, one type functions as the translation system, while the other is applied in a pre- or post-processing step. In this way, the shortcomings of the main translation method can be mitigated by supplementing it with the other.

Lagarda et al. (2009) used a rule-based translation system, but implemented an additional post-processing step that employed a statistical model to correct RBMT translation errors. On a high-perplexity corpus, this method was found to preserve the robustness of the RBMT system's capabilities, while correcting some of its errors in the post-processing step.

Meanwhile, Sun et al. (2010) applied a statistical model, trained on a monolingual source-language corpus, to source data. This created an intermediate language, which more closely resembled the structure of the target language. The modified data was then fed into an RBMT system. Results showed an improvement over the baseline RBMT-only translation system, especially as the SMT preprocessing system was trained on larger corpora, or on corpora more similar to the test set. This reinforces the importance of training data to the SMT system, and is something the present research took into consideration by randomizing the parallel corpus.

Finally, the last hybrid approach we will discuss is the type employed in the present research: a rule-based preprocessing step whose output is fed into a SMT system. Xu et al. (2009) recursively applied a set of precedence rules for reordering to a syntactic dependency tree in a top-down approach. The reordered output then served as input to a SMT system, where the reordering of source data notably led to an increase in word alignment quality and a decrease in decoding time for all five languages tested. Hong et al. (2009) included a preprocessing step to overcome null alignment issues in English-Korean translations. Researchers inserted pseudo words and performed basic syntactic reordering, transforming English sentences to make them more similar to Korean target sentences, in terms of length and word order. With reduced distortion and fertility, translation quality improved. Finally, Isozaki et al. (2010) applied a single preprocessing rule to English input for English-Japanese translations: for each syntactic constituent, move its syntactic head from constituent-initial position to final position. This rule was applied to not only verbs and adjectives as in the previous two methods, but also to function words such as prepositions. This approach greatly reduced word error rates, and showed a slight improvement in translation quality, as measured by automatic evaluation metrics.

While translation quality improvements were seen in systems with both pre- and post-processing configurations, the present research chose to implement a preprocessing step, rather than a post-processing step, in the hopes of limiting SMT decoding complexity. In the translation of divergent language pairs, the need for reordering adds considerable complexity to the translation process, in some cases more than doubling decoding time when long-distance reordering is performed (Bisazza & Federico 2013). Approaches with a postprocessing step translate

unmodified data, then attempt to correct errors in the translations. But, rather than relying on the translation system to spend extra cycles on reordering, then fixing detected errors, we determined a more productive goal would be to first modify the input to reduce the translation process to a more monotonic effort.

2.2 Reordering

While previous approaches have clearly illustrated the need for reordering when translating divergent language pairs, these approaches have largely fallen into two categories that, while serving as background for the present study, render the approaches insufficient for Korean to English MT.

First, many studies focus on language pairs where the source language is SVO and the target language is SOV. This is the case for the other rule-based preprocessing, stat-based translation hybrid approaches discussed in Section 2.1. Hong et. al (2009), when translating from English to Korean, inserted pseudo words to compensate for morpho-syntactic differences in the languages. However, translating in the other direction would seemingly then require the application of rules for *deletion* on Korean text to better match the target English structure. This idea of applying rules for deletion is explored in the present research. Similarly, Xu et al. (2009) used a precedence reordering approach on English sentences to mimic Korean structure. Their handwritten rules, while flexible, were created to only perform global, or long-distance, reordering. The present study does not aim to focus on either global or local reordering by themselves, as both are necessary to better bring Korean functional morphemes in line with English sentence structure.

Additionally, other approaches do not focus on the morphological complexity of the source language. This is either because neither source nor target language is agglutinative, in which case, the word can safely be considered as the smallest meaning-bearing unit of translation. Alternatively, it is because the research attempts to translate a *less* morphologically complex language to a *more* morphologically rich language. In this case, compensation for missing morphemes is often the main focus. Ramanathan et al. (2009) generated inflections and case markers for translation from English to Hindi, which uses case markers and morphological suffixes to confer meaning, while Li et al. (2010) looks at postposition generation in Chinese-Korean MT. The generation of missing information is not particularly applicable in Korean-English MT, where, rather, extraneous information must be filtered out or dealt with.

Perhaps most relevant to the present research, however, is El-Kahlout & Oflazer (2010), which exploits morphological information to improve English-Turkish translation quality. The study dealt with only lexical morphemes, rather than their surface forms which can vary due to vowel harmony restrictions. For statistical simplification and clarity, this is an idea the present research will also employ. The results also led researchers to question whether root words and functional morphemes should be handled using the same underlying mechanism, as they found morpheme ordering to be a much more constrained process. For this reason, this paper focuses only on functional morpheme reordering, and leaves the reordering of content root words like verbs and nouns to external components, in this case, the SMT translation system Moses.

2.3 Korean to English MT

To the best of our knowledge, Korean-English MT research is quite sparse, with most research efforts instead focusing on English-Korean MT, due to its myriad applications for those interested in this particular language pair. Several studies on English-Korean MT were discussed in previous sections, and will not be repeated here.

Though limited, Korean-English MT is not entirely without precedent. Kim et al. (2010) used syntactic chunks in an Example-Based MT system to overcome problems associated with inserting or deleting words. In their approach, the chunks consisted of one content word with its surrounding related function words, such as articles, making them, essentially, syntactically-related *n*-gram phrases. Results showed an improvement in the translation of otherwise 'untranslatable' tokens, as well as an increase in reordering flexibility, compared to arbitrary *n*-grams that cross chunk boundaries. This approach showed promising results as far as local reordering, and but does not obviously handle long distance reordering. Nor can it properly account for intervening modifiers that come between a content word and any adjacent function words. Function words also become dependent on content words for their translation, and vice versa, which has the potential to exacerbate the data sparseness problem. Researchers mitigated this to some extent by defaulting to a word-based translation if no chunk alignment was found. This, however, seems to further complicate the already-complexified translation process.

The other main effort involving Korean to English machine translation, Na (2015), implemented two non-projective reordering parsers, which, after training

on parallel corpora, produced a hierarchical structure, termed a reordering tree, containing explicit reordering information for each source sentence. These reordered sentences were then translated using a phrase-based SMT system. To implement their proposed bottom-up reordering parser, researchers trained a discriminative model using a set of features based on word alignment data. This concept, and some of these features, namely context part-of-speech tags and words, serve as a base for the present research's feature set features, which will be discussed in further detail in Chapter 4. However, while both parsers saw increases in translation accuracy and quality, they focused on global reordering, which is not discriminative enough to capture functional morpheme movements. This previous research also used the chunk as the minimum unit of reordering for English, which could potentially cause similar problems as the chunk-based approach in Kim et al. (2010). Finally, hand-annotated sentences, which took six man-months to notate, served as training data for one of the parsers. As we mentioned in Section 2.1, the need for manual effort can be one of the main limitations of a MT system, as it leads to expensive costs for maintenance and extension. Thus, the present research endeavors to use an entirely automated process, as described in Chapter 4.

Chapter 3. Corpus Data and SMT System

In this chapter, we aim to provide an overview of the corpus itself (Section 3.1) and its preparation as input to both the statistical machine translation system and our own rule generation system (Section 3.2). We also briefly outline details of the statistical machine translation system itself (Section 3.3).

3.1 Background

The spoken-language parallel corpus was provided for research purposes by Samsung. It consists of 349,874 Korean-English sentence pairs, mainly from the travel domain. Idiomatic translations such as "What's the holdup?" or "Mark my word." are a common occurrence, and were one of the biggest shortcomings of using this corpus in our automated rule generation task. Our task relies heavily on trained word alignments, and idiomatic expressions, which are not a literal word-for-word translations, confound the alignment statistics with figurative wording.

The Korean portion of the corpus was pre-tagged for part-of-speech (POS) using a Sejong Treebank parser. The Sejong Treebank (Korean Language Institute 2012) is the largest constituent treebank in Korean, with approximately 45 POS tags in use in the treebank. A sample of an annotated Korean sentence from the parallel corpus is shown in Figure 3.

The English portion, on the other hand, provided similar syntactic dependency information, in a different format. Sentences were POS-tagged using the Penn Treebank (Marcus et al. 1993) tags. The parse of Figure 3's corresponding English translation is shown in Figure 4.

<id 1>
<sent 1>
영어/NNP|로/JKB
설명/NNG|하/XSV|기/ETN
어렵/VA|습니다/EF|./SF
</sent>
</id>

Figure 3. A sample Korean sentence from the parallel corpus

(ROOT (S (NP (PRP It)) (VP (VBZ 's) (ADJP (JJ hard)
(S (VP (TO to) (VP (VB explain) (NP (PRP it)) (PP (IN in)
(NP (NNP English)))))))) (. .)))

Figure 4. A sample English sentence from the parallel corpus

3.2 Preparation

Prior to its use in either the translation system Moses, or in the present research's reordering rule generation system, we perform a bit of cleanup on the corpus. The cleanup consists mainly of correcting incorrect parses and recovering lost sentence-final punctuation. The former adjustment is important to ensure consistent contexts for rule generation. Meanwhile, it is also necessary to fix cases in which sentence-final punctuation is provided for only one side of the bilingual corpus. This is because word alignment requires the corpus to be segmented at the sentence level, and we automatically perform this segmentation

Problem	Before	Change
Incorrect parse	<id 1> <sent 1> 이태원/NNP|에/JKB 가/VV|ㄹ까/EC 합/VV|니다/EF|./SF </sent> </id>	합/VV|니다/EF|./SF → 하/VV|ㅂ니다/EF|./SF
Sentence-final marking restoration	<id 1> <sent 1> 겨울/NNG|이/JKS 오/VV|았/EP|어/EC </sent> </id>	오/VV|았/EP|어/EC → 오/VV|았/EP|어/EF|./SF

Table 1. Examples of performed corpus cleanup

```
Korean:   영어|NNP 로|JKB 설명|NNG 하|XSV 기|ETN 어렵|VA 습니다|EF .|SF
English:  It|PRP 's|VBZ hard|JJ to|TO explain|VB it|PRP in|IN English|NNP .|.
```

Figure 5. The parallel corpus formatted for Moses input

based on final punctuation markers. Representative cleanup examples are provided in Table 1.

Following the cleanup, the corpus sentence order is randomized while maintaining sentence pair alignment. Its format is adjusted to match the format required by Moses (see Figure 5), then it is divided into three sets: training (90%), tuning (5%), and testing (5%). The original, unmodified sets serve as input to the word alignment tool GIZA++ as well as to our reordering system (Och & Ney 2003). All three sets, in both their original and reordered forms, serve as input to Moses. In particular, the training set is used to generate word alignments in GIZA++, determine functional morpheme contexts and reordering movements for rule generation in our system, and build the language model in Moses. The tuning

	Total sentences	Korean morphemes	English words
Train (90%)	314,852	2,938,056	2,432,893
Tune (5%)	17,534	163,307	134,862
Test (5%)	17,488	163,373	135,516
Total (100%)	349,874	3,264,736	2,703,271

Table 2. Training, tuning and testing set statistics

set is used by Moses to minimize error rates after training, though with such a large training corpus, tuning was found not have an effect on translation performance. Finally, translation performance is evaluated with the test set: before applying our reordering rules, the original data is input to establish a Moses baseline for comparison; after reordering, the modified data is used to measure the effect of our reordering rule application. Details of the three sets are provided in Table 2.

3.3 Moses

Moses (Koehn et al. 2007) is an open-source toolkit for phrase-based statistical machine translation. Details regarding its determination of translation probability are beyond the scope of this summary, but suffice to say, many configurable parameters are provided to manipulate the statistical calculations. The present research utilizes Moses as a translation system, following the application of the core of our research, the automatically-generated reordering rules for Korean functional morphemes. We retain many of the default settings, but we also modify several key parameters. In some of our test cases, we also modify lexicalized reordering parameters to disable or heavily penalize reordering in

Moses, or not. In the case where we aim to limit reordering, we specify *monotonicity* reordering orientation during training, and set *distortion-limit* to 0 during decoding.

Now that we have summarized the translation system, as well as its expected input, let us begin our in-depth look at the core of our research, the rule generation process, in the following chapter.

Chapter 4. Rule Generation

Up to this point, we have outlined previous research in machine translation in general, and reordering and Korean to English translation in particular. We also described the corpus data used in our research, as well as the off-the-shelf statistical MT system, Moses, that we will use as the actual translation decoder. In this chapter, however, we detail our proposed approach to improve translation quality over the baseline system, Moses: a preprocessing step where functional morphemes are reordered to better approximate English sentence structure. It is based not on whether the movement is short (local) or long (global), but rather on the *type* of morpheme that serves as trigger for the reordering process. In this way, we can apply a more fine-grained level of movement, one that can be either local or global, to the set of functional morphemes in Korean.

In broad terms, our method is as follows. In order to reorder the Korean functional morphemes to their proper position in English sentence structure, we first create a classification model to disambiguate their various meanings. This involves first using word pair alignments to determine the suggested translations for each applicable morpheme, then building feature sets containing context information, and finally analyzing the type of reordering movement necessary to best match English sentence structure. Then, using these context feature sets and their associated movement labels, we generate reordering rules to be applied to the original, unmodified Korean input sentences. The newly reordered sentences are then used in both training and testing. We describe the details of these processes in the following subsections, and the results of our efforts in Chapter 5.

POS tag	Description	Example
EC	Conjunctive EM	V -고 'V and'
EF	Final EM	V -(으)ㅂ시다 'Let's V'
EP	Prefinal EM	V -았/었 (past tense V)
ETM	Adnominalizing EM	V -(으)려는 N (N that intends to V)
ETN	Nominalizing EM	V -기 (nom. form of V)
JC	Conjunctive PR	N -(이)나 'N or'
JX	Auxiliary PR	N -만 'only N'
JKB	Adverbial CP	N -처럼 'like N'
JKG	Adnominal CP	N -의 'N's'
JKS	Subjective CP	N -이/가
JKO	Objective CP	N -을/를
JKC	Complemental CP	N -이/가
JKV	Vocative CP	N -(이)야
JKQ	Quotative CP	N -(이)라고

Table 3. Relevant part-of-speech tags, descriptions, and examples
EM: ending marker, PR: particle, CP: case particle
N: noun, V: verb

4.1 Corpus Processing

In an agglutinative language like Korean, the number of functional morphemes is quite extensive. The ones that this research is concerned with, particles and ending markers, are divided into 14 part-of-speech (POS) tags in the Sejong Treebank. A complete list of relevant tags, adapted from Choi & Palmer (2011), is shown in Table 3. As the table illustrates, there are three main types: case particles, particles, and ending markers. POS tags that start with E indicate endings, e.g. pre-final, final, or verbal conjunctions. JC and JX, respectively, mark nominal conjunctions like ‘and’, and auxiliary information such as sentence topic. Finally, the POS tags beginning with JK are the case particles. They

Eng → Kor	Where is the convention hall ? NULL ({ }) 연회장|NNG ({ 3 4 5 }) 은|JX ({ 2 }) 어디|NP ({ 1 }) 죠|EF ({ }) ?|SF ({ 6 })
Kor → Eng	연회장|NNG 은|JX 어디|NP 죠|EF ?|SF NULL ({ }) Where ({ 3 }) is ({ 2 }) the ({ }) convention ({ 1 }) hall ({ }) ? ({ 4 5 })
Joint*	3-1 2-2 1-3 1-4 1-5 4-6 5-6

Table 4. GIZA++ alignment pairs
* listed as Kor-Eng pairs

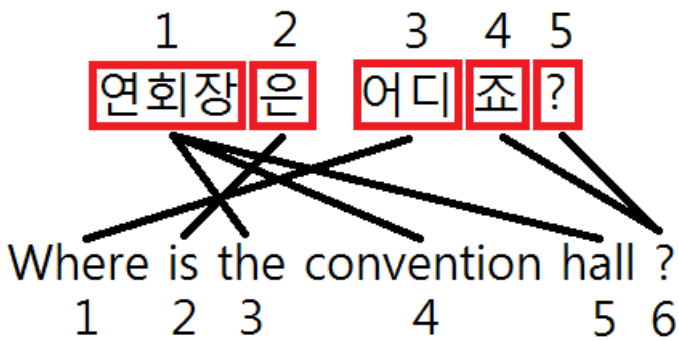

Figure 6. An illustration of the *grow-diag-final-and* joint alignment

represent grammar functions like sentence subject and object, source and destination of movement, and more.

4.1.1 Suggested Korean-English Alignments

As the first step in the rule generation process, we first use GIZA++ (Och & Ney 2003) on the original bilingual corpus to create a list of Korean-English alignment pairs. The Korean sentences were separated into individual morphemes, so that GIZA++, a word-level alignment tool, would essentially treat the

morphemes as words for the purposes of alignment determination. As a robust alignment tool, GIZA++ offers several different types of alignment models. We experimented with each of the nine types offered (grow, grow-diag, grow-diag-final, grow-diag-final-and, grow-final, intersect, srctotgt, tgttosrc, and union). It should be noted that in all cases, we utilized source and target POS tags as additional alignment factors. Using the GIZA++ *final* output files as the first inputs to our automated system, we found that *grow-diag-final-and* provides the highest number of usable alignment pairs, as determined by the pruning method described below. A sample *grow-diag-final-and* alignment pair is shown in Table 4, with a detailed illustration of its joint alignment in Figure 6.

As the original corpus was not reordered in any way, we expect that many of the alignment pairs would need to be excluded from further processing. Using the similar *grow-diag-final* alignment model, Hong et al. (2009) found that approximately 25% of words in Korean sentences and 21% of words in English sentences fail to align. These types of errors are not considered as invalid in the present research, however, as we plan to make deletion rules to compensate for null alignments. Instead, we are more concerned with *implausible* alignments, wherein the POS of the suggested GIZA++ translation is not a plausible match given the Korean POS. For example, when it is suggested that the conjunctive ending -고'and' be translated as the general noun 'dinosaur', we exclude this alignment pair from the pool of data to be used in the next step, feature set creation. Generally, these excluded POS tags represent numbers, verbs, nouns, pronouns, or punctuation, but a complete list of excluded English POS tags can be found in Table 5. We also exclude the alignment pair if the source-to-target and the target-to-source suggested translations do not agree. An exception to this is made if one

POS tag	Description	POS tag	Description
NN	noun, singular or mass	.	punctuation
NNS	noun, plural	CD	cardinal number
NNP	proper noun, singular	WRB	wh-adverb
NNPS	proper noun, plural	WP	wh-pronoun
PRP	personal pronoun	WDT	wh-determiner
VB	verb, base form	JJ	adjective
VBD	verb, past tense	JJR	adjective, comparative
VBN	verb, past participle	JJS	adjective, superlative
DT	determiner		

Table 5. English POS tags excluded from word alignment pairs

POS tag	Total count	Used count	Percent retained
EC	169,499	70,910	41.84
EF	291,336	117,590	40.36
EP	127,718	50,879	39.84
ETM	93,626	41,881	44.73
ETN	4,712	2,256	47.88
JC	3,823	2,974	**77.79**
JX	30,321	15,766	52.00
JKB	114,843	64,221	55.92
JKG	22,295	13,769	61.76
JKS	69,691	27396	39.31
JKO	80,301	38,103	47.45
JKC	2,685	813	*30.28*
JKV	65	31	47.69
JKQ	20	11	55.00
TOTAL	1,010,935	446,600	44.18

Table 6. Word pair retention rates per Korean POS tag (training set)

direction (either the source-to-target alignment, or the target-to-source alignment) indicates a null alignment, but the other has a plausible suggestion; in this case, the plausible translation is used. Overall, 44.28% of alignment pairs are deemed plausible, and used in the creation of context feature sets. Table 6 shows a per-POS tag breakdown of retention rates. JC tags, which represent nominative

Feature	Description
Context morphemes	Surrounding morphemes, at *pos*-2, *pos*-1, *pos*+1, *pos*+2
Context POSes	Surrounding part-of-speech tags, at *pos*-2, *pos*-1, *pos*+1, *pos*+2
Sentence type	Declarative, interrogative, exclamatory, imperative, propositional
Subject person type	1st, 2nd, 3rd, null
English translation	Suggested, based on GIZA++ word alignment pairs

Table 7. Feature set features and descriptions

conjunctions, have the highest rate of inclusion at 77.79%, while complement case marking morphemes (JKC) have the lowest rate of retention, at 30.28%.

4.1.2 Feature Sets

After eliminating implausible alignment pairs, approximately 446,000 pairs remained to be turned into feature sets, which are comprised of contextual information features from the Korean sentence as well as the suggested English translation. A complete list of features can be found in Table 7. The surrounding context morpheme and POS information, with the threshold window of +/- 2 word positions was adapted from Na (2015). Source sentence type (statement, question, etc.) was included to more accurately mimic English sentence structure, as English, unlike Korean, undergoes *wh*-word movement in interrogative type sentences. Subject person type was also incorporated into the feature set to see if it had an effect on rule generation; but, as only 69,691 of 314,852 (22.14%) training sentences had an explicit subject, this feature proved to be rather ineffective.

Generalization	Example
Vowel harmony neutralization	아/어 → 어 present tense marker
Vowel epenthesis after a coda	으면, 면 → 면 'if'; 이라서, 라서 → 라서 'because'
Formality neutralization	ㅂ니다, 에요, 야 → 야 marker after the copula 이/VCP

Table 8. Surface form morpheme generalizations with examples

Before any further analysis, the present research also generalizes morpheme forms, standardizing any morphemic allomorphs that arose due to vowel harmony restrictions, the presence or absence of coda consonants, or differing formality levels. This type of generalization was performed by El-Kahlout & Oflazer (2010), where researchers used Turkish lexical morphemes rather than their various surface forms in their English-Turkish MT research. Therefore, researchers determined that this reduction removed unnecessary fragmentation in statistical analysis, and presented a single, unified form when examining output. An example of our reductions is included in Table 8. After form generalization, we are left with 138 unique functional morpheme types.

4.1.2.1 Set Merging and Filtering

For each of the 138 general morpheme types, we count and combine all its feature sets with the same suggested translation. As one of the key points of this research is that functional morpheme translations vary based on context, it is no surprise that many types had more than one translation.

Looking at the merged feature sets, we realized that some bad alignments made it through our first round of *implausible POS* screenings. Thus, we employ

another round of filtering at this stage as well. Here, we aim to remove plausible but less frequent translations, by removing those with less than 50% of the highest translation's count, per type. We deemed this a more appropriate filtering method than simply choosing the top first or second quartile, as it places the cutoff threshold at 50% of the 'best' translation's count, rather than at 25% or 50% of the total count of all translations. Given many counts of approximately the same value, the quartile method will only select the top 25% or 50%, regardless of how similar the values are. For our particular task, we are more interested in selecting the top translations by count, regardless of the number selected. Selecting the top *X* translations, however, can lead to the selection of translations with much lower counts compared to the top-count translation. Thus, our selected method was employed. After merging and filtering, we are left with 268 unique context feature sets to turn into rules.

4.1.3 Reordering Movement

The next and final step in rule generation is determining the type of movement necessary to shift the Korean functional morpheme to its proper position in English sentence structure. In order to reorder the morpheme, we analyze the relative syntactic locations of each of the remaining feature set translations. To ensure that, in cases of more than one of the same lexical morpheme form per sentence, we analyze the correct morpheme-word pair, we actually insert this step back at the word alignment analysis stage, where each word pair's Korean morpheme and English word are represented as the *n*-th morpheme and *n*-th word

Movement type	Abbrev.	Representative example
Phrase Internal (Short)	PI	다른 학교에 → 에 다른 학교 another school-at → at another school 친구와 함께 → 와 친구 함께 friend-with together → with friend together
Clause Internal (but Phrase External)	CI	학교에 가면 → 면 학교에 가 school-at go-if → if school-at go
Clause External	CE	학교에 갔지만 → 학교에 갔, 지만 school-at went-but → school-at went, but
Other	O	topic, subject, object particles

Table 9. The major movement types within Korean sentence structure necessary to match English structure

PI Subtype	Morpheme movement	Example
Same Phrase, Nominal (SPN)	Before immediately preceding noun and its modifiers (adjectives, adnouns, etc).	Postposition to preposition
Same Phrase, Verbal (SPV)	Before immediately preceding verb and its modifiers (adverbs)	Adverbial -만 'only'

Table 10. Phrase Internal (PI) movement subtypes

in the input sentences. Our analysis revealed four major types of movement, which we term Phrase Internal (PI), Clause Internal (CI), Clause External (CE) and Other (O). These movement types are exemplified and defined in detail in Table 9.

4.1.3.1 Phrase Internal (PI)

Phrase Internal (PI) movement is defined as morpheme movement within a Korean syntactic phrase, particularly noun and verb phrases. Table 10 outlines the two subtypes, Same Phrase, Nominal (SPN) and Same Phrase, Verbal (SPV).

CI Subtype	Morpheme movement	Example
Same Clause, Independent of Sentence Type (SCI)	Front of the current clause	-면, -(ㄴ)다면 'if'
Same Clause, Dependent on Sentence Type (SCD)	Front of the current clause, before or after the subject (in non-question or question sentence types, respectively)	-어야 'should/must'

Table 11. Clause Internal (CI) movement subtypes

In the case of SPN, the morpheme to be reordered moves before the preceding noun and all its modifiers. A similar type of movement is done for SPV, except the movement now involves the preceding verb and its modifiers, rather than the preceding noun and its modifiers.

4.1.3.2 Clause Internal (but Phrase External) (CI)

Clause Internal (but Phrase External) (CI) movement is defined as morpheme movement outside the Korean syntactic phrase, but within the current clause. There are two movement subtypes, outlined in Table 11. The first is Same Clause, Independent of Sentence Type (SCI). This movement involves a shift to the beginning of the current clause. SCD is similar to the SCI, but due to question word fronting in English, this second type, which distinguishes between sentence type, is necessary. In this case, depending on whether the sentence is a question or not, the morpheme is reordered before or after the subject, respectively.

4.1.3.3 Clause External (CE)

Clause External (CE) movement is defined as morpheme movement

CE Subtype	Morpheme movement	Example
Next Clause, Independent of Sentence Type (NCI)	Front of the next clause	-지만 'but'

Table 12. Clause External (CE) movement subtypes

O Subtype	Morpheme movement	Example
Delete (DEL)	Delete	Topic, subject, object particles
Unmoving (UNM)	No movement	-나 'or' Default movement, if unspecified

Table 13. Other (O) movement subtypes

outside the current clause. Unlike Clause Internal movement, there were no movements found to be dependent on sentence type. Thus, only one subtype is presented, in Table 12. In Next Clause, Independent of Sentence Type (NCI), the morpheme is moved to the front of the next clause.

4.1.3.4 Other (O)

The Other (O) type of movement is, as its name suggests, comprised of movement types that do not fit into the other major categories. These include Deletion and Unmoving movements. Though the default reordering action is to leave the morpheme in its present position, in some cases, specifying Unmoving is necessary, for example, when another action, such as Delete, is the catch-all for that morpheme. The details are provided in Table 13.

Movement		POS tags	Morphemes
Type	**Subtype**		
PI	SPN	JKB, JKC, JX	과, 까지, 께, 대로, 도, 랑, 로, 로서, 마다, 만, 만큼, 밖에, 보다, 부터, 뿐, 서부터, 에, 에게, 에게서, 에서, 이, 처럼, 치고, 하고, 한테, 한테서
	SPV	EC, EF, ETM, JX	ㄴ지, ㄹ지, 러, 려는, 만, 은데, 지
CI	SCD	EC, EF, EP, ETM	게, 겠, 까, ㄴ가, 나, 는가, 도, ㄹ, ㄹ게, ㄹ까, ㄹ지, 래, 려면, 면, ㅂ니까, 세, 습니까, 시, 야, 어도, 어서, 어야, 예요, 을까, 을래, 을지, 자, 지
	SCI	EC, EF, EP, JKB	께, ㄴ다면, 는군, 는데, 니까, 다면, 다면서, ㄹ로, 라, 라서, 로부터, 면, 면서, ㅂ시오, 세, 시, 야, 어다, 어도, 에서, 오, 은데, 을걸, 자
CE	NCI	EC, JX	거나, 고, ㄴ가, ㄴ데, ㄴ지, 니까, 도, 만, 며, 므로, ㅂ니까, 습니까, 어서, 예요, 지만
O	DEL	EC, EF, EP, ETF, ETN, JKB, JKC, JKO, JKQ, JKS, JKV, JX	가, 거나, 거든, 게, 겠, 고, 고서, 고자, 구나, 군, 기, 께서, ㄴ, ㄴ가, ㄴ다, ㄴ다고, ㄴ다는, ㄴ데, ㄴ지, 나, 네, 느냐, 는, 는군, 는다, 는데, 는지, 니, 니다, 다, 다가, 다고, 다는, 다니, 답니다, 대, 더군, 던, 던데, 데, 도, 도록, 든, 든지, ㄹ, 라, 라고, 라는, 라도, 라면, 라서, 려고, 려는, 로써, 를, 리라고, ㅁ, ㅂ시다, 뿐, 서, 세, 셔, 습니까, 습니다, 시, 시다, 야, 어, 어도, 어라, 어야, 었, 었었, 에, 예요, 요, 은, 은데, 을, 을게, 을지, 음, 이, 잖아, 처럼, 치고
	UNM	EC, EF, JC, JKB, JKG, JKS, JX	거든, 게, 고자, 과, ㄴ, ㄴ가, ㄴ데, 나, 냐, 는다, 니, 든, ㄹ까, 랑, 려, 려고, 려면, 서, 습니까, 야, 요, 의, 지

Table 14. Morphemes and POS tags, per movement type

4.1.3.5 Functional Morpheme Summary per Movement Type

Table 14 provides a complete list of all analyzed functional morphemes and associated POS tags, per movement type and subtype.

The Phrase Internal (PI) major movement type consists mainly of adverbial, complemental, and auxiliary case particles, but also contains conjunctive, final, and adnominalizing ending markers. In the case of the case particles, which are of subtype Same Phrase, Nominal (SPN), this indicates movement from a post-nominal to pre-nominal position. The adverbial case particles listed here, such as −만 'only N', −마다 'every N' or −처럼 'like N', in fact modify nouns. Thus, while technically labeled adverbials, as far as English sentence structure, their positions shift around the Korean nouns they are attached to. Similarly, the ending markers found in the Same Phrase, Verbal (SPV) subtype, such as −지, which functions as post-verbal negation, and −러 'going to V', end up being analyzed as adverbs modifying a verb.

The Clause Internal (CI) movement type is composed primarily of ending markers, but some adverbial case particles are included as well. The ending markers all represent grammar functions that are expressed at the beginning of the clause in English structure, but are attached to the verb at the end of the clause in Korean. For example, −면 'if', −니까 'because' and −어야 'should/must' all shift to the beginning of the clause when translating from Korean to English. In the case of the former two, the movement is independent of sentence type. The latter's overall movement, however, depends on whether the sentence is a question, or not: "Should I eat lunch?" versus "I should eat lunch."

Clause External (CE) movement consists of only one subtype, Next Clause, Independent of Sentence Type (NCI), which is comprised of conjunctive ending markers like -고 'and' and -므로 'therefore', and one auxiliary case particle, -습니다. -습니다 is usually a final ending marker, but in this context, it appears with the adverbial -만 and is translated as 'but'.

The final type, Other (O), which actually represents either deletion or lack of movement, consists of the most morphemes overall. For reasons explained at the end of this subsection, many of the morphemes found here are also found in other movement types as well. However, in certain contexts, the morpheme is to be deleted or left alone. As mentioned in Section 4.1.3.4, Unmoving is the default movement type -- that is, unless movement is prescribed by a rule match, the morpheme is not moved. The morphemes listed here often have "otherwise delete" catch-all rules which replace the default movement type. Therefore, if the morpheme is to remain in place, Unmoving must be specified with an additional rule indicating the UNM context. Additionally, in our reordering system, each analyzed morpheme must have at least one rule. Sometimes that rule is merely UNM. This is the case, for example, for all the conjunctive case particles which attach to nouns: -과 'and', -나 'or', and -랑 'and'.

DEL-type morphemes necessitate a separate explanation. DEL-type morphemes are made up of the largest number of functional morphemes by unique type. These morphemes mainly represent concepts that have no translation at all in English, such as subject, object, and topic particles. However, they also include morphemes that have no direct translation in English, such as -라도. This functional morpheme attaches to a noun, and indicates that the noun is not the

most preferred option, but it will suffice in the given situation. It could be translated as "at least", as in "Let's at least eat bread." But in general, and in our corpus in particular, it is rarely translated this way. Thus, our system marks it for deletion. We found several additional instances of Korean morphemes that express concepts or feelings and do not have direct translations in English. Therefore, it was important that our system capture the need for deletion in the usual case where there is no translation at all, but also when there is no translation realized. This is done through the use of our context feature sets. In this way, we are able to cover cases where there is one translation for the morpheme in certain contexts, and a different translation, or perhaps no translation at all, in others. For this reason, Table 14 contains a number of morphemes that are repeated in several movement types. For example, the case particle -에 is typically translated as 'to' or 'at', so it is listed in the movement type SPN. However, when preceded by certain NNG-marked nouns, like 다음 'next' or 전 'before', the 'to/at' translation is not realized, and our rules correctly capture that the morpheme should be deleted. Therefore, it is also listed in DEL.

4.2 Rule Creation

Now that we have both the rule context feature set and the rule reordering movement result, as determined in the previous section, let us look at the actual method of rule creation.

For each unique lexical morpheme type, each context feature set including its suggested translation becomes the left-hand side of a reordering rule. Its

Rule	Description
① 만 → E(PM2(VV),PM1(ETN))%SPV	When *pos*-1 POS is ETN and *pos*-2 POS is VV, move before the verb
② 만 → E(PM2(VV),PM1(EC))%SPV	When *pos*-1 POS is EC and *pos*-2 POS is VV, move before the verb
③ 만 → E(MM1(습니다))%NCI	When *pos*-1 morpheme is 습니다, move to the front of the next clause
④ 만 → E()%SPN	Otherwise, move before the preceding noun and its modifiers

Table 15. Reordering rules and descriptions for 만 'only'

Rule	Description
① 뿐 → E(MP1(ᄋᆡ),PP1(VCP))%SPN	When the *pos*+1 morpheme is ᄋᆡ and the *pos*+1 POS is VCP, move before the preceding noun and its modifiers
② 뿐 → E()%DEL	Otherwise, delete

Table 16. Reordering rules and descriptions for 뿐 'only'

movement type becomes the right-hand side. Table 15 shows an example rule set for the auxiliary particle, 만 'only'. These rules work in conjunction with rules for the auxiliary particle, 뿐 'only', shown in Table 16. As they both translate to 'only', per the rules, only one, 만, is retained when both appear together in a Korean sentence.

Overall, 268 reordering rules are created for 138 unique lexical morpheme types, with 61 types necessitating the creation of more than one rule for disambiguation. A complete list of these automatically generated rules, as determined by the methods described previously in this chapter, can be found in Appendix A.

	Train (Percent Reordered)	Test (Percent Reordered)
Reordered	805,642 (NA)	43,491 (NA)
Relevant	1,010,935 (79.69)	52,920 (82.18)
Total	2,938,056 (27.42)	163,373 (26.62)

Table 17. Morphemes reordered, training and test sets

4.3 Input Preprocessing

Prior to language model training and translation in Moses, each sentence in the training, tuning, and test sets must be preprocessed. Each functional morpheme is checked against the features of any applicable rules (lefthand side). If the necessary context exists, the morpheme is reordered per the associated movement type (righthand side). This preprocessing step is detailed in the subsections below. The number of reordered morphemes for the training and test sets are shown in Table 17. Here, *total* represents the total morphemes, while *relevant* refers to only functional morphemes.

4.3.1 Rule Matching

The algorithm used for rule matching is outlined in Table 18, but a short description will be provided here as well.

Basically, for each relevant Korean morpheme in the sentence, we look up the list of its associated rules. For each rule, if all the features in the rule match the morpheme's context, then the corresponding reordering action and the total count of rule features are added to a list as a pair. After repeating this for each rule, we have a list of possible reordering actions paired with the strength of the

action's match (i.e. the number of features the morpheme had to match to meet the rule). Noting the number of matched features is important because many of the morphemes, such as 뿐 'only' in Table 16, have a 'default' reordering action, often DEL. So any 뿐|JX morpheme that matches the first rule will also match the default rule. We must also consider the case where more than one action has the same match count. In this case, we prioritize movements in the order of CE > CI > PI > O, with DEL given lowest priority. Since the reordering action list is sorted, we can stop looking for equal counts when the count value changes. Finally, we add the selected action to the sentence-level reordering action list.

Algorithm 1. Rule Matching

rule_list: the list of reordering rules for each morpheme-POS pair
action: the reordering action to be taken
count: the number of features in the rule that had to be matched
action_list: a list of possible reordering actions and their matched feature count
reorder_map: a map of morphemes and their selected reordering action

```
for each morph in korean_sent do

    for each rule in rule_list do
        valid ← true
        for each feature in rule do
            if feature not found in morph context then
                valid ← false
                break
            end if
        end for
        if valid == true then
            action_list ← add (action, count)
        end if
    end for

    sort-descending action_list by count

    action ← action_list[0].action

    for i = 0 to action_list.size - 1 do
        if action_list[i].count == action_list[i+1].count then
            if action_list[i].action == DEL then
                action ← action_list[i+1].action
            else
                action ← select action[i] or action[i+1], prioritizing CE>CI>PI>O
            end if
        else
            break
        end if
    end for

    reorder_ map ← add (morph, action)

end for
```

Table 18. Rule matching algorithm

4.3.2 Morpheme Reordering

Now that the rule matching step described in Section 4.3.1 is complete and we have a list of morphemes to be reordered along with their corresponding reordering actions, we can begin to reorder them as needed. While the entire Rule Matching algorithm was outlined in Table 18, only the algorithm for Same Clause, Independent of Sentence Type (SCI) is shown in Table 19, for space considerations.

Overall, we begin by looking at each morpheme in the original Korean sentence. If it does not have an associated reordering action, we simply append the morpheme to the end of the new, reordered sentence, which we are building from left to right. We chose to build left to right because most reordering actions shift the morphemes leftward, due to Korean being a head-final, post-positional language, while English is head-initial and prepositional. In the cases (e.g. NCI movement) where we need to shift to the right, the morpheme is flagged and shifted later, after we have appended other morphemes later in the sentence. On the other hand, if the morpheme does have an associated reordering action, we reorder it accordingly. SCI, SCD, SPN, SPV all have similar actions: we begin at our position, which is the current end of the reordered sentence being built, and search backwards until we reach a stop condition, or the beginning of the sentence. In the case of SCD, we perform the same algorithm as SCI in Table 19, but we adjust the morpheme insertion based on sentence type. Meanwhile, morphemes with SPN or SPV movements are inserted after the first morpheme reached that is a non-noun (including its modifiers) or a non-verb (with its modifiers), respectively. Finally, morphemes with UNM movements are directly appended to the current position, while DEL morphemes are simply not added to the reordered sentence.

Algorithm 2. Morpheme Reordering

```
reordered_sent:    the reordered sentence, built from left to right
reorder_map:    a map of morphemes and their selected reordering action
VERB_SET:    a list of verbal POS tags
STOP_SET:    a list of verb and punctuation POS tags

for each morph in orig_sent do
    if morph not in reorder_map then
        append morph to end of reordered_sent
    else
        action ← reorder_map[morph]
        switch(action):
            case 'SCI':
                stopAt ← -1;    passedVerbAt ← -1; passedVerb ← false
                pos ← current reordered_sent position
                for q = pos - 1 to 0 do
                    compare_morph ← reordered_sent.morphemes[q]
                    if compare_morph.pos in VERB_SET then
                        passedVerb ← true
                    else if passedVerb == true then
                        passedVerbAt ← q+1
                        break
                    end if
                end for

                if passedVerbAt > 0 then
                    for r = passedVerbAt - 1 to 0 do
                        compare_morph ← reordered_sent.morphemes[r]
                        if compare_morph in STOP_SET then
                            stopAt ← r
                            break
                        end if
                    end for
                end if

                if stopAt != -1 then
                    insert morph at reordered_sent[stopAt+1]
                else
                    insert morph at reordered_sent[0]
                end if
            end case
        end switch
    end if
end for
```

Table 19. Rule reordering algorithm

4.4 Examples

To further illustrate the preprocessing method described throughout this chapter, let's look at a brief example, shown in Table 20. For the 'original' translation, Moses was trained using the unmodified bilingual corpus, with the default Moses settings (e.g. distortion limit = 6) and *msd* (monotone, swap, discontinuous) reordering orientation. For the 'new' translation, Moses was retrained using the same data, but after it had been reordered via our preprocessing step. We also show two reordering set-ups: 1) the default settings and *msd*

ORIGINAL KOREAN INPUT to MOSES
우리|NP 는|JX 곧|MAG 호찌민|NNP 에|JKB 도착|NNG 하|XSV ㄹ|ETM 것|NNB 이|VCP ㅂ니다|EF .|SF

GOLD STANDARD TRANSLATION
We 'll soon be arriving at Ho Chi Minh.

ORIGINAL ENGLISH OUTPUT from MOSES, default reordering settings
We Ho Chi Minh shortly 778 will arrive in.

APPLICABLE REORDERING RULES
는 → E()%DEL
에 → E()%SPN
ㄹ → E(MP1(것),PP1(NNB),MP2(이),PP2(VCP))%SCD
습니다 → E(MP1(.),PP1(SF),MP2(NULL),PP2(NULL),ST(S))%DEL

REORDERED KOREAN INPUT to MOSES
우리|NP ㄹ|ETM 곧|MAG 에|JKB 호찌민|NNP 도착|NNG 하|XSV 것|NNB 이|VCP .|SF

NEW ENGLISH OUTPUT from MOSES, restricted reordering
We will soon at Ho Chi Min arrive.

NEW ENGLISH OUTPUT from MOSES, less restricted reordering, with defaults
We will arrive soon at Ho Chi Min .

Table 20. Example of reordering rule application

orientation, which was used for the ‘original’ translation, and 2) another configuration wherein additional configuration options were provided to limit the reordering done by Moses (distortion limit=0, *monotonicity* reordering orientation). It should be noted that we also retrained and tested ‘original’ Moses using the reordering limitation options, but the output was found to be the same as when reordering was not limited.

Neither the ‘original’ nor the ‘new’ translation exactly matches the Gold Standard. However, compared to the ‘original’, our preprocessing approach makes several crucial improvements:

- the preposition ‘in’ correctly comes before the noun 'Ho Chi Minh'
- ‘will’ correctly appears at the beginning of the sentence, following the subject
- the incorrect word pair alignment which led to the appearance of ‘778’ has been amended

If we compare the ‘original’ unreordered output to the ‘new’ reordered but less restricted output, it can be seen that while our preprocessing step only deals with functional morphemes and leaves all content word reordering to Moses, the changes we make increase Moses’ ability to reorder longer distances. In the ‘original’ output, the verb remains in its Korean position at the end of the sentence. After our changes, however, given the same default options, it is able to be reordered to its proper English location.

Two further examples of the effects of our reordering system are shown in Table 21 and Table 22. As before, the morpheme reorderings produced by our system cause marked changes in Moses' output.

ORIGINAL KOREAN INPUT to MOSES
좋|VA 습니다|EF .|SF 이|MM 서류|NNG 를|JKO 좀|MAG 작성|NNG 하|XSV 아|EC 주|VX 시|EP 어요|EF .|SF

GOLD STANDARD TRANSLATION
Fine , you have to fill out this form .

ORIGINAL ENGLISH OUTPUT from MOSES, default reordering settings
good . form fill this out .

REORDERED KOREAN INPUT to MOSES
좋|VA .|SF 시|EP 이|MM 서류|NNG 좀|MAG 작성|NNG 하|XSV 주|VX .|SF

NEW ENGLISH OUTPUT from MOSES, restricted reordering
good . please this form fill out .

NEW ENGLISH OUTPUT from MOSES, less restricted reordering, with defaults
good . please fill out this form .

Table 21. Another example of reordering rule application

ORIGINAL KOREAN INPUT to MOSES
대전|NNP 에서|JKB 특급|NNG 열차|NNG 에서|JKB 보통|MAG 열차|NNG 로|JKB 갈아타|VV 시|EP ㅂ시오|EF .|SF

GOLD STANDARD TRANSLATION
Change from the express to the local at Daejeon .

ORIGINAL ENGLISH OUTPUT from MOSES, default reordering settings
local at Daejeon you transfer to please .

REORDERED KOREAN INPUT to MOSES
에서|JKB 대전|NNP 에서|JKB 특급|NNG 열차|NNG 로|JKB 보통|MAG 열차|NNG 갈아타|VV ㅂ시오|EF .|SF

NEW ENGLISH OUTPUT from MOSES, restricted reordering
from Daejeon express train by local train transfer Please .

NEW ENGLISH OUTPUT from MOSES, less restricted reordering, with defaults
at Daejeon from express train to local train Please transfer .

Table 22. Final example of reordering rule application

Now that we have explained the methodology behind our preprocessing approach in detail, we will present the overall results of our efforts in the next chapter, Chapter 5.

Chapter 5. Results

For the following preprocessing experiments, we use monotone reordering settings in Moses, which limits the amount of reordering Moses does on its own. This is done to show the effect our changes have, without conflation. The baseline we use for comparison, however, was also executed with the default settings, which allow limited reordering on the part of the SMT system.

The translation quality of the output was measured using BLEU score (Papineni 2002). BLEU score is the benchmark automatic evaluation metric for machine translation. It compares several factors between Reference (or Gold) and Candidate translations, including length (precision), word choice (recall), and word order (edit distance). Given the precision p_n of *n*-grams up to size N, the length of the Candidate translation *c*, and the length of the Reference translation *r*, the BLEU score is calculated as follows:

$$\text{BLEU} = \text{BP} \cdot \exp\left(\sum_{n=1}^{N} \log p_n\right) \quad (2)$$

$$\text{BP} = \min(1, e^{1-r/c}) \quad (3)$$

This formula assigns the highest score to a Candidate translation that most closely matches the Reference translation, as score deductions are applied for brevity, lengthiness, and omitted words, among other factors.

BLEU score results for several test setups are listed in Table 23. BASE indicates the original, unreordered data, decoded in Moses using monotone parameters. For the BASE+PI, BASE+CI, BASE+CE, and BASE+O trials, we applied the methods described in Section 4.3 to match and reorder the morphemes

Movement Type	BLEU score
BASE	31.89
BASE+PI	32.60
BASE+CI	32.51
BASE+CE	32.07
BASE+O	32.47
BASE+ALL	**33.01**
AVG (CHG) / MAX (CHG)	32.53 (.64) / 33.01 (1.12)

Table 23. BLEU score by movement type, restricted reordering in Moses

based on context, for a single major movement type (including all its subtypes). In the BASE+ALL trial, we applied the rules for all movement types. Among the individual movement types, the application of the phrase internal (PI) movement rules provided the biggest BLEU score increase (0.71 absolute, 2.2% relative), followed by Clause Internal (CI) (0.62 absolute, 1.9% relative), Other (O) (0.58 absolute, 1.8% relative), and finally, Clause External (CE) (0.18 absolute, 0.9% relative). The combination of all four types, however, shows the most significant increase, from 31.89 to 33.01 (1.12 absolute, 3.5% relative).

Though Clause External movements are, by definition, movements outside the clause, in actuality, the movements necessary to shift from Korean word order to English word order are not that long. This is because the morphemes are originally located at the end of the clause, so the movement required to shift the morpheme to the beginning of the next clause in English sentence structure is relatively short. Thus, the lower increase in BLEU score could be accounted for by the limited shift in morpheme location. Meanwhile, Phrase Internal and Clause Internal movements are actually longer in

Movement Type	BLEU score
BASE	31.89
BASE, less restricted reordering	32.03
BASE+Global only	32.58
BASE+ALL, excluding Other	32.77

Table 24. BLEU score for combination types, restricted reordering in Moses

distance, and therefore effect changes that would not be accounted for in the baseline.

Table 24 lists the results for several more trials. In these experiments, we tested other combinations of reordering rules and Moses parameters. BASE once again represents the original, unreordered input using a monotone Moses decoder for translation. In the second case, with less restricted reordering, we used default Moses parameters, enabling limited reordering to be done by the decoder. In the next case, we tested global-level reordering by applying the rules for both Clause Internal and Clause External movements, and excluded Phrase Internal and Other. Finally, in the last trial, we applied all movement rules except Delete. These results highlight two main points of our reordering system. First, they indicate that even the individual movement type with the lowest BLEU score increase (CE, 0.18 absolute, 0.9% relative) shows an improvement over what Moses itself can do, given its default settings, without the addition of our preprocessing step. Second, they show that with our reordering system, local (PI) and global (CI+CE) reordering is nearly equivalent (0.71 and 0.69, respectively) when decoder-time reordering is restricted.

	Training data (sentences)	Base BLEU	Final BLEU	Absolute increase	Relative increase (%)
Na (2015)	601,896	19.58	20.45	0.87	4.25
Our approach	314,852	32.03	33.51	1.48	4.42

Table 25. Comparison to recent Korean-English MT research, Na (2015), unrestricted reordering in Moses

It should also be noted that the application of the rules for multiple movement types is not quite an additive process. For example, the Other type, added to BASE alone, shows an increase of 0.58. Meanwhile, its omission only shows a drop of 0.24 compared to its inclusion. This implies the rules affect each other, and that there exists a maximal order in which to apply each rule. In the Rule Matching step described in Section 4.3.1, we attempted to maximize this order by prioritizing movement type selection. These results reflect the highest value obtained.

As previously mentioned, to the best of our knowledge, recent Korean-English MT research is quite limited. The most recent endeavor is Na (2015), who implemented two reordering parsers which, in a preprocessing step, produced a Korean-English reordering tree containing explicit reordering information, for each Korean sentence. The reordered sentences were then translated using Moses.

In our final experiment, we look at one of Na (2015)'s parsers, the bottom-up parser. This parser used a combination of word- and POS-based features, as well as transition boundary features to classify the data. In our case, we applied the preprocessing rules for all movement types to reorder the data. For a more balanced comparison between our methods, we followed Na (2015)'s Moses configuration. Using default distortion and lexicalized reordering settings, we

obtained another, larger BLEU score increase, compared to monotone settings. Table 25 shows a comparison of the two approaches. It should be noted, however, that Na (2015)'s preprocessing approach mainly covers global reordering, which, as previously stated, SMT systems alone are insufficient at dealing with. Meanwhile, our approach looks at a more fine-grained reordering, both local and global, for functional morphemes only. In particular, the reordering of content root words such as verbs and nouns is left to the SMT system, Moses. Therefore, while a basic comparison of the two approaches is possible, a 1:1 analysis would not be entirely appropriate.

Chapter 6. Conclusion

Overall, the present research implemented reordering at the morpheme level, which is more relevant for agglutinative languages like Korean, and to the best of our knowledge, the present study constitutes the first attempt to do so. Our hybrid approach focused on reordering Korean functional morphemes to match English sentence structure in a rule-based preprocessing step, and left the translation step to an external SMT system, Moses.

We performed several experiments where we allowed the application of different reordering rules based on the major movement types, PI, CI, CE, and O, both individually and all together. These experiments all limited the reordering done by Moses itself, in order to not conflate our reordering system results with Moses' own reordering processes. According to the standard evaluation metric, BLEU score, the results for individual movement types were ordered as follows: PI (0.71 absolute increase, 2.2% relative), CI (0.62 absolute increase, 1.9% relative), O (0.58 absolute increase, 1.8% relative), and CE (0.18 absolute increase, 0.9% relative). The greatest BLEU score increase, however, was achieved with the application of all rules for all movement types (1.12 absolute, 3.5% relative). A second set of experiments showed that 1) even the lowest increase (CE movement) was found to be an improvement over the baseline when default reordering parameters were applied to Moses, and 2) the improvements for local and global reordering were nearly equivalent (0.71 and .069, respectively). Finally, when we lifted the reordering restrictions to the default values, we saw an even greater BLEU score increase of 1.48 (4.42%). This indicates that the lexicalized reordering offered by Moses is still providing a benefit in addition to the more fine-

grained reordering done by our preprocessing step. This is not against expectation, however, as our preprocessing step only deals with functional morphemes, while content words like nouns and verbs are left unreordered. Additionally, as seen in the examples in Section 4.4, there are cases when the reordering done by our preprocessing step actually enables Moses to perform necessary content word reordering that it otherwise could not.

The final feature of the present study worth noting is that all steps, from GIZA++'s word alignment and our own rule generation and reordering preprocessing step, to the translation itself in Moses, are done automatically. Though the training set size is large, at over 300,000 sentences, our method enabled us to forgo hand annotation, which is costly in terms of both time and money. Our system is also set up such that, with a few adjustments, it could be reused with other corpora or even expanded and adapted to other language pairs.

The next step, in future research, would be to expand the system to automatically generating rules for grammatical functions that consist of more than one morpheme. The present research only deals with endings and particles, but these functional morphemes frequently appear in the same context as certain other morphemes. It might prove beneficial to merge or otherwise handle them together before the reordering stage.

References

Bar-Hillel, Y. (1951). The present state of research on mechanical translation. American Documentation, 2(4), 229-237.

Berger, A. L., Brown, P. F., Della Pietra, S. A., Della Pietra, V. J., Kehler, A. S., & Mercer, R. L. (1996). U.S. Patent No. 5,510,981. Washington, DC: U.S. Patent and Trademark Office.

Bisazza, A., & Federico, M. (2013, August). Efficient solutions for word reordering in German-English phrase-based statistical machine translation. In Proceedings of the Eighth Workshop on Statistical Machine Translation (pp. 440-451).

Choi, J.D., & Palmer, M. (2011, October). Statistical dependency parsing in Korean: From corpus generation to automatic parsing. In The Second Workshop on Statistical Parsing of Morphologically Rich Languages (pp. 1-11). Association for Computational Linguistics.

Collins, M., Koehn, P., & Kučerová, I. (2005, June). Clause restructuring for statistical machine translation. In Proceedings of the 43rd annual meeting on association for computational linguistics (pp. 531-540). Association for Computational Linguistics.

Dryer, M. S. (1991). SVO languages and the OV: VO typology. Journal of Linguistics, 27(02), 443-482.

Dryer, M. S. (1992). The Greenbergian word order correlations. Language, 81-138.

El-Kahlout, I. D., & Oflazer, K. (2010). Exploiting morphology and local word reordering in English-to-Turkish phrase-based statistical machine translation. IEEE Transactions on Audio, Speech, and Language Processing, 18(6), 1313-1322.

Genzel, D. (2010, August). Automatically learning source-side reordering rules for large scale machine translation. In Proceedings of the 23rd international conference on computational linguistics (pp. 376-384). Association for Computational Linguistics.

Herrmann, T., Niehues, J., & Waibel, A. (2013, June). Combining Word Reordering Methods on different Linguistic Abstraction Levels for Statistical Machine Translation. In Proceedings of the Seventh Workshop on Syntax, Semantics and Structure in Statistical Translation, Altanta, Georgia, USA, June. Association for Computational Linguistics.

Hong, G., Lee, S. W., & Rim, H. C. (2009, August). Bridging morpho-syntactic gap between source and target sentences for English-Korean statistical machine translation. In Proceedings of the ACL-IJCNLP 2009 Conference Short Papers (pp. 233-236). Association for Computational Linguistics.

Hutchins, J. (2007). Machine translation: A concise history. Computer aided translation: Theory and practice.

Isozaki, H., Sudoh, K., Tsukada, H., & Duh, K. (2010, July). Head finalization: A simple reordering rule for sov languages. In Proceedings of the Joint Fifth Workshop on Statistical Machine Translation and MetricsMATR (pp. 244-251). Association for Computational Linguistics.

Kim J. D., Brown R. D., & Carbonell, J. G. (2010) Chunk-Based EBMT. In: Proceedings of the 14th workshop of the European Association for Machine Translation (EAMT-2010)

Koehn, P., Och, F. J., & Marcu, D. (2003, May). Statistical phrase-based translation. In Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1 (pp. 48-54). Association for Computational Linguistics.

Koehn, P., Hoang, H., Birch, A., Callison-Burch, C., Federico, M., Bertoldi, N., ... & Dyer, C. (2007, June). Moses: Open source toolkit for statistical machine translation. In Proceedings of the 45th annual meeting of the ACL on interactive poster and demonstration sessions (pp. 177-180). Association for Computational Linguistics.

Korean Language Institute (2012). Sejong treebank. http://www.sejong.or.kr.

Lagarda, A. L., Alabau, V., Casacuberta, F., Silva, R., & Diaz-de-Liano, E. (2009, May). Statistical post-editing of a rule-based machine translation system. In Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics, Companion Volume: Short Papers (pp. 217-220). Association for Computational Linguistics.

Lee, Y. S., Zhao, B., & Luo, X. (2010, August). Constituent reordering and syntax models for English-to-Japanese statistical machine translation. In Proceedings of the 23rd international conference on computational linguistics (pp. 626-634). Association for Computational Linguistics.

Li, J. J., Kim, J., & Lee, J. H. (2010). Transferring Syntactic Relations of Subject-Verb-Object Pattern in Chinese-to-Korean SMT.

Marcus, M. P., Marcinkiewicz, M. A., & Santorini, B. (1993). Building a large annotated corpus of English: The Penn Treebank. Computational linguistics, 19(2), 313-330.

Na, Hwidong. (2015). Non-projective Parsing for Pre-ordering Statistical Machine Translation (Unpublished doctoral dissertation). Pohang University of Science and Technology, Pohang, Korea.

Och, F. J., & Ney, H. (2003). A systematic comparison of various statistical alignment models. Computational linguistics, 29(1), 19-51.

Papineni, K., Roukos, S., Ward, T., & Zhu, W. J. (2002, July). BLEU: a method for automatic evaluation of machine translation. In Proceedings of the 40th annual meeting on association for computational linguistics (pp. 311-318). Association for Computational Linguistics.

Ramanathan, A., Choudhary, H., Ghosh, A., & Bhattacharyya, P. (2009). Case markers and morphology: Addressing the crux of the fluency problem in english-hindi smt. In Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP (pp. 800–808). Suntec, Singapore, August. Association for Computational Linguistics.

Rottmann, K., & Vogel, S. (2007). Word reordering in statistical machine translation with a POS-based distortion model. Proc. of TMI, 171-180.

Sohn, H. M. (2001). The Korean Language. Cambridge University Press.

Sun, Y., O'Brien, S., O'Hagan, M., & Hollowood, F. (2010). A novel statistical pre-processing model for rule-based machine translation system. Proceedings of EAMT, 8pp.

Vogel, S. (2003, October). SMT decoder dissected: Word reordering. In Natural Language Processing and Knowledge Engineering, 2003. Proceedings. 2003 International Conference on (pp. 561-566). IEEE.

Wang, C., Collins, M., & Koehn, P. (2007). Chinese syntactic reordering for statistical machine translation. In Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (pp. 737-745).

Wu, D. (1996, June). A polynomial-time algorithm for statistical machine translation. In Proceedings of the 34th annual meeting on Association for Computational Linguistics (pp. 152-158). Association for Computational Linguistics.

Xu, P., Kang, J., Ringgaard, M., & Och, F. (2009, May). Using a dependency parser to improve SMT for subject-object-verb languages. In Proceedings of human language technologies: The 2009 annual conference of the North American chapter of the association for computational linguistics (pp. 245-253). Association for Computational Linguistics.

Appendix A: Rules

POS Tag: EC

거나 → E(PM1(VV))%NCI
거나 → E()%DEL
게 → E(MM1(그렇),PM1(VA))%UNM
게 → E()%DEL
고 → E(MP1(있),PP1(VX))%DEL
고 → E(MP1(싶),PP1(VX))%DEL
고 → E(MP1(말),PP1(VX))%DEL
고 → E(MP1(나),PP1(VX))%DEL
고 → E()%NCI
고서 → E(PM1(VV))%DEL
고자 → E(PM2(NNG),MM1(하),PM1(XSV),MP1(하),PP1(VX))%UNM
고자 → E(PM2(NNG),MM1(하),MP1(하),PP1(VX))%DEL
까 → E(PM2(VV),MM1(르),PM1(ETM),MP1(생각),PP1(NNG))%SCD
ㄴ가 → E(PM2(NP),MM1(이),PM1(VCP))%DEL
ㄴ가 → E(PM2(NNG),MM1(이),PM1(VCP),PP2(NNG),ST(Q))%NCI
ㄴ가 → E(MM2(르),PM2(ETM),MM1(거),PM1(NNB),ST(Q))%NCI
ㄴ가 → E(MM1(이),PM1(VCP))%UNM
ㄴ다고 → E()%DEL
ㄴ다면 → E()%SCI
ㄴ데 → E(MM1(하),MP1(,),PP1(SP))%NCI
ㄴ데 → E(MM1(이),PM1(VCP))%DEL
ㄴ지 →
E(MM2(는지),PM2(EC),MM1(어떻),PM1(VA),MP1(말),PP1(NNG),MP2(하),PP2(XSV),ST(Q))%NCI
ㄴ지 → E(MM1(이),PM1(VCP),ST(Q))%SPV
ㄴ지 → E()%DEL
나 → E()%DEL
느냐 → E(PM1(VV))%DEL
는데 → E()%DEL
는지 → E()%DEL
니 → E()%DEL
니까 → E()%NCI
다 → E()%DEL
다가 → E()%DEL
다고 → E()%DEL
다니 → E()%DEL
다면 → E()%SCI
도 → E(PM2(VV),MM1(어),PM1(EC),MP1(되),PP1(VV))%SCD
도 → E(PM1(VV),MP1(되),PP1(VV))%SCD
도 → E(MM1(들어가),PM1(VV),MP1(되),PP1(VV),ST(Q))%NCI

도 → E()%DEL
도록 → E()%DEL
든 → E(MM2(언제),PM2(NP),MM1(이),PM1(VCP))%UNM
든 → E(MM1(이),PM1(VCP))%DEL
든지 → E(PM2(NP),MM1(이),PM1(VCP))%DEL
르까 →
E(MM2(에),PM2(JKB),MM1(가),PM1(VV),MP1(하),PP1(VV),MP2(습니다),PP2(EF))%UNM
르지 →
E(MM2(어야),PM2(EC),MM1(하),PM1(VX),MP1(모르),PP1(VV))%SPV
르지 → E()%SCD
라 → E(PM2(NNG),MM1(이),PM1(VCP),PP1(NNG))%SCI
라 → E(MM1(이),PM1(VCP))%DEL
라고 →
E(MM2(어),PM2(EC),MM1(보),PM1(VX),MP1(제안),PP1(NNG),MP2(하),PP2(XSV))%DEL
라도 → E(MM1(이),PM1(VCP),PP1(VV))%DEL
라면 → E(MM1(이),PM1(VCP))%DEL
라서 → E(PM2(NNG),MM1(이),PM1(VCP))%SCI
라서 → E(MM1(이),PM1(VCP),PP1(NNG))%DEL
러 → E(PM2(NNG),MM1(하),PM1(XSV))%SPV
러 → E(PM1(VV),MM1(가),PP1(VV))%SPV
려 → E(PM2(NNG),MM1(하),PM1(XSV),MP1(하),PP1(VX),PP2(EF))%UNM
려고 → E(PM2(NNG))%UNM
려고 → E(MP1(하),PP1(VX))%DEL
려면 → E(PM1(VV))%SCD
려면 → E(PM1(VV),MP1(어느),PP1(MM),PP2(NNG))%UNM
리라고 → E(MP1(생각),PP1(NNG),MP2(하),PP2(XSV))%DEL
며 → E()%NCI
면 → E(PM2(EC),PM1(VV),MP1(되),PP1(VV),PP2(EF),ST(Q))%SCD
면 → E()%SCI
면서 → E()%SCI
므로 → E(PM1(VV))%NCI
ㅂ니까 → E(PM2(NNG),MM1(이),PM1(VCP),MP1(,),PP1(SP))%NCI
ㅂ니까 → E(MM2(거),PM2(NNB),MM1(이),PM1(VCP),MP1(,),PP1(SP))%NCI
서 → E()%DEL
습니까 → E(MM2(시),PM2(EP),MM1(겠),PM1(EP),MP1(,),PP1(SP))%NCI
습니다 → E(MM1(하),MP1(만),PP1(JX))%DEL
야 → E(PM2(JKB),PM1(VV),MP1(하),PP1(VX),PP2(EF),ST(Q))%SCD
야 → E(MP1(하),PP1(VX))%DEL
야 → E(MP1(하),PP1(VX),PP2(EF))%SCI
야 →
E(MM2(게),PM2(EC),MM1(가),PM1(VV),MP1(하),PP1(VX),MP2(나),PP2(EF),ST(Q))%UNM

어 → E(PP1(VX))%DEL
어다 →
E(MM2(에),PM2(JKB),MM1(데리),PM1(VV),MP1(주),PP1(VX),MP2(시),PP2(EP))%SCI
어도 → E(PP1(VV))%SCI
어도 → E(PM2(NNG),MP1(되),PP1(VV))%SCD
어도 → E(MP1(되),PP1(VV))%SCD
어도 → E()%DEL
어라 →
E(MM2(지),PM2(EC),MM1(말),PM1(VX),MP1(.),PP1(SF),MP2(NULL),PP2(NULL))%DEL
어서 → E()%NCI
어야 → E(PP1(VX),PP2(EF))%SCD
어야 → E()%DEL
예요 →
E(MM2(거),PM2(NNB),MM1(이),PM1(VCP),MP1(,),PP1(SP),ST(Q))%NCI
예요 → E(MM1(이),PM1(VCP),MP1(,),PP1(SP))%DEL
요 → E(MM1(이),PM1(VCP),PP1(SF),MP2(NULL),PP2(NULL))%DEL
요 →
E(MM1(이),PM1(VCP),MP1(.),PP1(SF),MP2(NULL),PP2(NULL))%UNM
은데 → E()%DEL
을지 → E(MP1(도),PP1(JX),MP2(모르),PP2(VV))%SCD
을지 → E(MM2(수),PM2(NNB),MM1(있),PM1(VV),PP1(VV),PP2(EP))%SCD
을지 → E(MM1(있),PM1(VV),PP1(VV))%DEL
자 → E()%SCI
지 → E(PP1(VX))%SPV
지만 → E()%NCI

POS Tag: EF
거든 →
E(PM2(NNG),MM1(이),MP1(.),PM1(VCP),PP1(SF),MP2(NULL),PP2(NULL),ST(S))%UNM
거든 → E(MP1(.),PP1(SF),MP2(NULL),PP2(NULL),ST(S))%DEL
게 → E(MM1(르),PM1(ETM),PP1(SF),MP2(NULL),PP2(NULL))%SCD
고 → E(PP1(SF),MP2(NULL),PP2(NULL))%DEL
구나 → E(MP1(.),PP1(SF),MP2(NULL),PP2(NULL),ST(S))%DEL
군 → E(PP1(SF),MP2(NULL),PP2(NULL))%DEL
까 → E(MM1(르),PM1(ETM),PP1(SF),MP2(NULL),PP2(NULL))%SCD
께 → E(MM1(르),PM1(ETM),PP1(SF),MP2(NULL),PP2(NULL))%SCI
ㄴ가 → E(PP1(SF),MP2(NULL),PP2(NULL))%DEL
ㄴ가 → E(MM1(이),PM1(VCP),PP1(SF),MP2(NULL),PP2(NULL))%SCD
ㄴ가 → E(MM1(멀),PM1(VA),PP1(SF),MP2(NULL),PP2(NULL))%UNM

ㄴ다 → E(MP1(.),PP1(SF),MP2(NULL),PP2(NULL),ST(S))%DEL
ㄴ데 → E(MM1(이),PM1(VCP),PP1(SF),MP2(NULL),PP2(NULL))%UNM
나 → E(PP1(SF),MP2(NULL),PP2(NULL))%DEL
나 →
E(PM2(NNG),MM1(있),PM1(VV),PP1(SF),MP2(NULL),PP2(NULL))%SCD
나 → E(PM1(VV),PP1(SF),MP2(NULL),PP2(NULL))%SCD
나 →
E(MM2(수),PM2(NNB),MM1(있),PM1(VV),PP1(SF),MP2(NULL),PP2(NULL))
%SCD
냐 → E(MM1(시),PM1(EP),PP1(SF),MP2(NULL),PP2(NULL))%UNM
네 → E(PP1(SF),MP2(NULL),PP2(NULL))%DEL
느냐 → E(PP1(SF),MP2(NULL),PP2(NULL))%DEL
는가 → E(PP1(SF),MP2(NULL),PP2(NULL))%SCD
는군 → E(PM1(VV),MP1(.),PP1(SF),MP2(NULL),PP2(NULL),ST(S))%SCI
는군 → E(MP1(.),PP1(SF),MP2(NULL),PP2(NULL),ST(S))%DEL
는다 → E(PP1(SF),MP2(NULL),PP2(NULL))%DEL
는다 →
E(MM2(지),PM2(EC),MM1(않),PM1(VX),PP1(SF),MP2(NULL),PP2(NULL))%
UNM
는데 → E(PP1(SF),MP2(NULL),PP2(NULL))%DEL
는데 →
E(MM2(고),PM2(EC),MM1(있),PM1(VX),PP1(SF),MP2(NULL),PP2(NULL))%
SCI
는지 → E(PM1(EP),PP1(SF),MP2(NULL),PP2(NULL))%DEL
니 →
E(MM2(좋),PM2(VA),MM1(았),PM1(EP),PP1(SF),MP2(NULL),PP2(NULL))%
UNM
니까 → E(PP1(SF),MP2(NULL),PP2(NULL))%SCI
니다 → E(PP1(SF),MP2(NULL),PP2(NULL))%DEL
다 → E(PM1(EP),PP1(SF),MP2(NULL),PP2(NULL))%DEL
다고 → E(PP1(SF),MP2(NULL),PP2(NULL))%DEL
다니 → E(PM1(VV),PP1(SF),MP2(NULL),PP2(NULL))%DEL
다면서 → E(PP1(SF),MP2(NULL),PP2(NULL))%SCI
답니다 → E(PP1(SF),MP2(NULL),PP2(NULL))%DEL
대 → E(PP1(SF),MP2(NULL),PP2(NULL))%DEL
더군 → E(PP1(SF),MP2(NULL),PP2(NULL))%DEL
던데 → E(MP1(.),PP1(SF),MP2(NULL),PP2(NULL),ST(S))%DEL
데 → E(PP1(SF),MP2(NULL),PP2(NULL))%DEL
르게 →
E(MM2(어),PM2(EC),MM1(보),PM1(VX),PP1(SF),MP2(NULL),PP2(NULL))%
SCD
르까 → E(PP1(SF),MP2(NULL),PP2(NULL))%SCD
라 → E(MM1(어),PM1(EC),PP1(SF),MP2(NULL),PP2(NULL))%DEL
라고 → E(PP1(SF),MP2(NULL),PP2(NULL))%DEL

래 →
E(MM2(시),PM2(EP),MM1(ㄹ),PM1(ETM),PP1(SF),MP2(NULL),PP2(NULL))
%SCD
래 → E(MM1(ㄹ),PM1(ETM),PP1(SF),MP2(NULL),PP2(NULL))%SCD
려고 →
E(PM2(NNG),MM1(하),PM1(XSV),PP1(SF),MP2(NULL),PP2(NULL))%UNM
ㅂ니까 →
E(MM1(이),PM1(VCP),MP1(?),PP1(SF),MP2(NULL),PP2(NULL),ST(Q))%SCD
ㅂ시다 → E(MP1(.),PP1(SF),MP2(NULL),PP2(NULL),ST(S))%DEL
ㅂ시오 →
E(MM2(주),PM2(VX),MM1(시),PM1(EP),PP1(SF),MP2(NULL),PP2(NULL))%S
CI
서 → E(MM1(어),PM1(EC),PP1(SF),MP2(NULL),PP2(NULL))%DEL
습니까 →
E(PM2(NNG),MM1(있),PM1(VV),MP1(?),PP1(SF),MP2(NULL),PP2(NULL),ST
(Q))%SCD
습니까 →
E(MM2(얼마나),PM2(MAG),MM1(걸리),PM1(VV),PP1(SF),MP2(NULL),PP2(
NULL))%UNM
습니까 →
E(MM2(수),PM2(NNB),MM1(있),PM1(VV),MP1(?),PP1(SF),MP2(NULL),PP2(
NULL),ST(Q))%SCD
습니까 →
E(MM1(있),PM1(VV),MP1(?),PP1(SF),MP2(NULL),PP2(NULL),ST(Q))%DEL
습니다 → E(MP1(.),PP1(SF),MP2(NULL),PP2(NULL),ST(S))%DEL
시다 → E(PM1(VV),PP1(SF),MP2(NULL),PP2(NULL))%DEL
야 → E(MM1(이),PM1(VCP),PP1(SF),MP2(NULL),PP2(NULL))%DEL
어 → E(PP1(SF),MP2(NULL),PP2(NULL))%DEL
어라 → E(MP1(.),PP1(SF),MP2(NULL),PP2(NULL),ST(S))%DEL
어서 →
E(MM2(고),PM2(EC),MM1(싶),PM1(VX),PP1(SF),MP2(NULL),PP2(NULL))%
SCD
어서 → E(MM1(있),PM1(VV),PP1(SF),MP2(NULL),PP2(NULL))%SCD
예요 → E(MM1(이),PM1(VCP),PP1(SF),MP2(NULL),PP2(NULL))%SCD
예요 → E(MM1(이),PM1(VCP),PP1(SF),MP2(NULL),PP2(NULL))%SCD
오 → E(MM1(시),PM1(EP),MP1(.),PP1(SF),MP2(NULL),PP2(NULL))%SCI
요 → E(PP1(SF),MP2(NULL),PP2(NULL))%DEL
은데 → E(PP1(SF),MP2(NULL),PP2(NULL))%DEL
은데 →
E(MM2(고),PM2(EC),MM1(싶),PM1(VX),PP1(SF),MP2(NULL),PP2(NULL))%
SPV
은데 →
E(MM2(것),PM2(NNB),MM1(같),PM1(VA),PP1(SF),MP2(NULL),PP2(NULL))
%SCI

을걸 →
E(MM2(고),PM2(EC),MM1(싶),PM1(VX),PP1(SF),MP2(NULL),PP2(NULL))%SCI
을게 → E(PP1(SF),MP2(NULL),PP2(NULL))%DEL
을까 →
E(MM2(수),PM2(NNB),MM1(있),PM1(VV),PP1(SF),MP2(NULL),PP2(NULL))%SCD
을래 → E(PP1(SF),MP2(NULL),PP2(NULL))%SCD
을래 →
E(MM2(지),PM2(EC),MM1(않),PM1(VX),PP1(SF),MP2(NULL),PP2(NULL))%SCD
자 → E(PP1(SF),MP2(NULL),PP2(NULL))%SCD
잖아 → E(PP1(SF),MP2(NULL),PP2(NULL))%DEL
지 → E(PP1(SF),MP2(NULL),PP2(NULL))%UNM
지 → E(MM1(이),PM1(VCP),PP1(SF),MP2(NULL),PP2(NULL))%SCD

POS Tag: EP
겠 → E(PP1(EF),PP2(SF))%DEL
겠 → E(MP1(습니다),PP1(EF),MP2(.),PP2(SF))%SCD
겠 →
E(MM2(주),PM2(VX),MM1(시),PM1(EP),PP1(EF),MP2(?),PP2(SF),ST(Q))%SCD
겠 →
E(MM2(주),PM2(VX),MM1(시),PM1(EP),MP1(어),PP1(EF),MP2(?),PP2(SF),ST(Q))%SCD
겠 → E(MM1(시),PM1(EP),PP1(EF),MP2(?),PP2(SF),ST(Q))%SCD
세 → E(PM1(VV),MP1(요),PP1(EF),PP2(SF))%DEL
세 → E(MM1(주),PM1(VX),MP1(요),PP1(EF),PP2(SF))%SCI
세 → E(MM1(있),PM1(VV),MP1(요),PP1(EF),MP2(?),PP2(SF))%SCD
셔 → E(MP1(야),PP1(EC),MP2(하),PP2(VX))%DEL
시 → E(ST(S))%DEL
시 → E(PM2(EC),MM1(주),PM1(VX),MP1(어),PP1(EF),MP2(.),PP2(SF))%SCI
시 → E(MP1(습니까),PP1(EF),PP2(SF))%DEL
시 →
E(MM1(있),PM1(VV),MP1(습니까),PP1(EF),MP2(?),PP2(SF),ST(Q))%SCD
었 → E(PP1(EF),PP2(SF))%DEL
었었 → E(MM1(하),PP1(EF),PP2(SF))%DEL

POS Tag: ETM
ㄴ → E()%DEL
ㄴ다는 → E()%DEL

는 → E()%DEL
다는 → E()%DEL
던 → E()%DEL
ㄹ → E(PM1(VV),PP1(NNB))%SCD
ㄹ → E(MP1(수),PP1(NNB),MP2(있),PP2(VV))%SCD
ㄹ → E(MP1(것),PP1(NNB),MP2(이),PP2(VCP))%SCD
ㄹ → E()%DEL
라는 → E(MM1(이),PM1(VCP))%DEL
려는 → E(PM1(XSV),PP1(NNG))%SPV
려는 → E(PM1(VV),PP1(NNG))%SPV
려는 → E()%DEL
은 → E()%DEL

POS Tag: ETN
기 → E()%DEL
음 → E()%DEL
ㅁ → E()%DEL

POS Tag: JC
과 → E(PM1(NNG),PP1(NNG))%UNM
나 → E(PM1(NNG),PP1(NNG))%UNM
랑 → E(PM1(NNG),PP1(NNG))%UNM

POS Tag: JKB
과 → E()%SPN
께 → E(PP1(NNG))%SPN
ㄹ로 →
E(MM2(NULL),PM2(NULL),MM1(이것),PM1(NP),MP1(하),PP1(VV))%SCI
랑 → E()%SPN
로 → E()%SPN
로부터 → E(MM2(NULL),PM2(NULL),PM1(NNP))%SCI
로서 → E(PM1(NNG))%SPN
로써 → E()%DEL
만큼 → E()%SPN
보다 → E()%SPN
서 → E(MM1(여기),PM1(NP))%UNM
서부터 → E(MM2(NULL),PM2(NULL),MM1(여기),PM1(NP))%SPN
에 → E()%SPN

에 → E(MM1(다음),PM1(NNG))%DEL
에 → E(MM1(전),PM1(NNG))%DEL
에게 → E()%SPN
에게서 → E()%SPN
에서 → E(PM1(NNG))%SPN
에서 → E(MM2(NULL),PM2(NULL))%SCI
처럼 → E(PM2(ETM),MM1(것),PM1(NNB))%DEL
처럼 → E()%SPN
하고 → E()%SPN
한테 → E()%SPN
한테서 → E()%SPN

POS Tag: JKC
가 → E(PM1(NNG),MP1(되),PP1(VV))%DEL
가 → E()%DEL
이 → E(PM1(NNG),MP1(아니),PP1(VCN),PP2(EF))%SPN
이 → E()%DEL

POS Tag: JKG
의 → E(PM1(NP),PP1(NNG))%UNM

POS Tag: JKO
르 → E()%DEL
를 → E()%DEL
을 → E()%DEL

POS Tag: JKQ
고 → E()%DEL
라고 → E()%DEL

POS Tag: JKS
가 → E()%DEL
께서 → E()%DEL
서 → E(MM1(혼자),PM1(NNG))%UNM
이 → E()%DEL

POS Tag: JKV
야 → E()%DEL

POS Tag: JX
까지 → E()%SPN
ㄴ → E(MM2(NULL),PM2(NULL),PM1(NP))%UNM
ㄴ → E()%DEL
나 → E()%DEL
는 → E()%DEL
대로 → E(PM1(NNG))%SPN
도 → E()%SPN
마다 → E(PM1(NNG))%SPN
만 → E(PM2(VV),PM1(ETN))%SPV
만 → E(PM2(VV),PM1(EC))%SPV
만 → E(MM1(습니다))%NCI
만 → E()%SPN
밖에 → E()%SPN
부터 → E()%SPN
뿐 → E(MP1(이),PP1(VCP))%SPN
뿐 → E()%DEL
요 → E(PP1(SF),MP2(NULL),PP2(NULL))%DEL
은 → E()%DEL
치고 → E(MM2(NULL),PM2(NULL),PM1(NNG),PP1(MAG))%SPN
치고 → E()%DEL

Abstract in Korean

기계 번역 연구에서 극복해야 하는 중요한 문제 중 하나는, 한국어-영어와 같이 어순이 서로 다른[SOV-SVO] 언어 쌍을 어떻게 재배열하여 처리할 것인가에 있다. 일반적으로 SMT(Statistical Machine Translation: 통계적 기계 번역) 모델에서는 왜곡 벌점(distortion penalty)이 부과되므로, 원거리 어순 재배열이 충분히 수행되지 못한다. 반면, 규칙 기반 시스템은 개발 및 유지에 많은 시간과 비용을 요구한다. 본 연구에서는 한국어의 영어 번역을 위한 새로운 혼합형 접근법(hybrid approach)을 제안한다. 단어 중심의 접근법을 시도한 선행 연구들과는 달리, 본 연구에서는 언어 쌍 번역의 기본 단위로 '형태소'를 고려한다.

본 연구는, 각 한국어 형태소의 특징적인 문맥 정보에 기반하여, 한국어 기능 형태소의 모호성(ambiguity)을 해결하는 분류 모델을 개발하는 데서 시작한다. 이후, 해당 분류 모델의 자동 생성 규칙을 사전 처리 단계에 적용하여 한국어 형태소를 영어의 어순으로 재배열한다. 마지막으로, SMT 시스템인 Moses를 사용하여 이를 영문으로 번역한다.

문맥 정보에 기반하여 한국어 형태소를 재배열한 병렬 코퍼스(parallel corpus)를 Moses에 재교육한 결과, 전반적인 번역 품질이 향상되었다. Moses 자체의 어휘화 재배열(lexicalized reordering)을 비활성화하고 해당 모델만을 적용하였을 때, Moses의 어휘화 재배열만을 사용했을 때보다 3.5% 증가한 BLEU 점수를 얻을

수 있었다. Moses의 어휘화 재배열까지 모두 활성화하는 경우, BLEU 점수가 4.42%로 더 크게 증가하는 것으로 관찰되었다. 본 연구는 각 형태소의 특징적 문맥 정보를 통해, 형태소 수준의 더 정교한 기계 번역이 가능함을 보여주었다는 데 의의가 있다.