BIDIRECTIONAL SEGMENTATION FOR ENGLISH-KOREAN MACHINE TRANSLATION

by

Youngchan Kim
B.Sc. (Hons.), University of Toronto, 2009

A THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF
MASTER OF SCIENCE
in the
School of Computing Science
Faculty of Applied Science

SIMON FRASER UNIVERSITY
Spring 2012

APPROVAL

Name: Youngchan Kim

Degree: Master of Science

Title of thesis: Bidirectional Segmentation for English-Korean Machine Translation

Examining Committee: Dr. Bob Hadley
Chair

Dr. Anoop Sarkar, Associate Professor, School of Computing Science
Simon Fraser University
Senior Supervisor

Dr. Veronica Dahl, Professor, School of Computing Science
Simon Fraser University
Supervisor

Dr. Fred Popowich, Professor, School of Computing Science
Simon Fraser University
Examiner

Date Approved: April 2, 2012

Partial Copyright Licence

Abstract

Unlike English or Spanish, which has each word clearly segmented, morphologically rich languages, such as Korean, do not have clear optimal word boundaries for machine translation (MT). Previous work has shown that segmenting such languages by incorporating information available from parallel corpus can improve MT results. In this thesis we show that this can be improved further by segmenting both source and target languages and present improvement in BLEU scores from 3.13 to 3.46 for English-Korean translation.

To my family and Anna

Acknowledgments

Foremost, I would like to thank my senior supervisor, Dr. Anoop Sarkar, for his guidance, support, and supervision throughout my master's program at Simon Fraser University. He gave me invaluable insights that shaped the ideas and approaches to this research.

I would like to thank my supervisor, Dr. Veronica Dahl, and examiner, Dr. Fred Popowich, for taking their time to review this thesis. I also would like to thank Dr. Bob Hadley for chairing my thesis defense.

Additionally, I would like to thank all my friends, lab mates, classmates, faculty members, and support staff from School of Computing Science.

Last but not least, I would like to thank my family and my girlfriend, Anna, for their strong support. It would have been impossible to finish this thesis without them.

Contents

Approval ii

Abstract iii

Dedication iv

Acknowledgments v

Contents vi

List of Tables viii

List of Figures ix

1 Introduction 1

1.1 Morphologically Complex Languages 1

1.2 Morphologically Complex Languages and SMT 2

1.3 Morpheme Segmentation for SMT 4

1.4 SMT for Korean 4

1.4.1 Related Work 5

1.5 BLEU Scores 6

1.6 Thesis Outline 9

2 Related Work 10

2.1 Monolingual Segmentation - Morfessor 10

2.2 Bilingual Segmentation 12

2.2.1 Background 12

2.2.2 Model 12

2.2.3 Morpheme Translation Model 14

2.2.4 Word Boundary Generation Model 15

2.2.5 Distortion Model 15

2.2.6 Length Penalty 15

2.2.7 Inference 16

2.3 Chapter Summary 17

3 Models **20**

3.1 Monolingual Model 21

3.2 Bilingual Model 22

3.2.1 Variational Bayes 24

3.3 Length Penalty 24

3.4 Results in Chinese-English and Korean-English Translation 26

3.5 Bidirectional Segmentation 28

3.6 Chapter Summary 29

4 Experiments and Results **31**

4.1 Log Scale in Bilingual Model 31

4.2 Data 32

4.3 Experimental Setup 33

4.3.1 Moses 33

4.3.2 Stem and Suffix 33

4.3.3 Data Set 33

4.3.4 Evaluation Methods 34

4.3.5 Merging 35

4.4 Results 36

4.5 Chapter Summary 38

5 Conclusion **39**

5.1 Future Works 40

References **41**

List of Tables

2.1 Features in log-linear distortion model firing for the transition from *te*:suffix:1 to *tsvet*:root:3 in the example sentence pair in Figure 1a of ((Naradowsky and Toutanova, 2011). Table from ((Naradowsky and Toutanova, 2011). 16

3.1 BLEU score results for Chinese-English and Korean-English experiments and F-score of segmentation compared against Chinese Treebank standard. Table from ((Chung and Gildea, 2009). 27

4.1 Comparison of different merge methods. 36

4.2 Sample segmented sentences from Rochester Korean-English data. Spaces surrounded by '+' represent newly added segmentations. For monolingual model, we used the one with maximum suffix length of 10. The original data is in UTF-8 but is romanized here for convenience. 37

4.3 BLEU score results for different methods in English-Korean machine translation. Numbers in brackets represent maximum length of suffix used when training the segmenter. 37

List of Figures

1.1 Morpholoical analysis of a Russian word ‘*antipravitelstvennyi*’ (antigovernment). . 2

1.2 Example of a Korean sentence that does not include subject. 5

2.1 The hierarchical segmentations of ‘straightforwardness’. Figure from ((Creutz and Lagus, 2005). 11

2.2 Bulgarian forms of ‘red’ ((Naradowsky and Toutanova, 2011) 12

2.3 A graphical depiction of the model generating the transliteration of a Bulgarian word ‘*chervenite*’. Trigram dependencies and some incoming/outgoing arcs have been omitted for clarity. Figure from ((Naradowsky and Toutanova, 2011). 13

2.4 Comparison of morpheme segmentation F-scores in between Morfessor ((Creutz and Lagus, 2005), S&B ((Snyder and Barzilay, 2008), Poon et. al ((Poon, Cherry, and Toutanova, 2009), and N&T ((Naradowsky and Toutanova, 2011) for Hebrew and Arabic bible data ((Snyder and Barzilay, 2008), and Arabic Tree Bank. Figures from ((Naradowsky and Toutanova, 2011). 18

2.5 Comparison of Alignment Error Rate (AER) in between C&G ((Chung and Gildea, 2009), WDHMM((He, 2007), and N&T ((Naradowsky and Toutanova, 2011). Figure from ((Naradowsky and Toutanova, 2011). 19

3.1 This figure shows a Korean sentence $\mathbf{e} = e_1, e_2 = \mathbf{s} \circ c_1 \dots c_4$ where $\mathbf{s} = (0, 0, 1, 1)$ and an English sentence $\mathbf{f} = f_1, f_2$. The segmentation between c_3 and c_4 means that c_1, c_2, c_3 forms the Korean word $e1$ and c_4 forms the next word e_2. These Korean words e_1 and e_2 are generated by the English words f_1 and f_2, respectively. 21

3.2 Distribution of token length for Chinese and Korean. ‘ref’ is the empirical distribution from supervised segmentation. Figure from ((Chung and Gildea, 2009). 25

3.3 Example of incorrect alignment between an English word and morpheme-segmented Korean word. '+' was used to mark the morpheme segmentation. 28

3.4 Overview of the bidirectional segmentation model. 29

4.1 Suffix break-down of a word '$c_1c_2c_3c_4c_5$' with the maximum suffix length specified as 3. 34

Chapter 1

Introduction

The main idea of statistical machine translation (SMT) is that we can translate between languages in an unsupervised manner by using large parallel texts and the statistics behind them. Because this idea of SMT first started from translating between English and similarly-structured languages, such as Spanish or French, a large number of SMT techniques that we have are originally developed towards translating from and into these languages. These traditional SMT techniques that are built towards English and similar languages tend to assume the word, which are marked by blank spaces around them, as the basic unit of analysis. This may be problematic when translating into or from the languages that are dissimilar to English. In this chapter, we will discuss how this may cause problems in morphologically-complex languages.

1.1 Morphologically Complex Languages

In linguistics, a morpheme is the smallest semantically meaningful unit in a language. For example, in English, a word '*cooked*' contains two morphemes: one is '*cook*' and the other is '*ed*', which is used to express a past tense. Morphological complexity refers to the degree to which languages use bound morphemes, mainly to express grammatical or derivational relations, and each language consists of different levels of morphological complexity. In terms of morphological complexity, English is typologically classified as an isolating language. On average, English has a relatively low number of morphemes per word when compared to many other languages and mainly relies on the word order to express syntactic relations Although we sometimes find English words that consist of multiple morphemes, such as '*cooked*', it is true that we can also find many words that consist of a single morpheme, such as '*bottle*'.

On the other hand, languages like Turkish, Russian, and Korean are considered as morphologically complex languages, which are also called synthetic languages. These languages can express various information in a word with multiple morphemes in it. For example, a Russian word '*antipravitelstvennyi*' (antigoverment) consists of six different morphemes, as can be seen from Figure 1.1. Although some of the morphemes in this word do not have an equivalent word in English, each morpheme has some meaning.

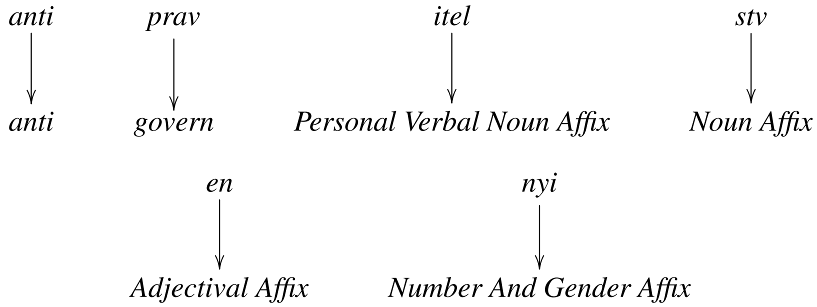

Figure 1.1: Morpholoical analysis of a Russian word '*antipravitelstvennyi*' (antigovernment).

1.2 Morphologically Complex Languages and SMT

Most word-based machine translation systems treat words bounded by blank space as atomic units. This means that they treat words '*sing*', '*sings*', '*singer*', and '*singers*' as totally different words despite the fact that they all share a common morpheme '*sing*'. Although this is not a major problem when translating between languages with similar morphological complexity and features, this may be highly problematic when translating between languages with different morphological complexity.

(Clifton, 2010) summarizes several challenges of morphologically complex languages for SMT:

- In the field of natural language processing, an n-gram is a contiguous sequence of n items from a given sequence of text. Morphologically complex languages tend to have freer word orders, since word order is not as syntactically salient. Hence, traditional n-gram statistics may be less informative.

- Words in morphologically complex languages are frequently composed of multiple morphemes inflecting a single or compound word stem. While there may be a limited stem and morphological lexicon, since these are productively combinable, this can lead to a combinatorial explosion of surface forms. Thus, any given surface form may occur quite infrequently in a text, leading to a significant data sparsity problem. This increases the number of out-of-vocabulary words in a morphologically complex text as well as makes text word occurrence statistics less robust.

- Translation between a morphologically rich language and one with little morphology suffers from the problem of source-target asymmetry. From a non-inflecting source language that expresses most syntactic relations lexically or through word order, the MT model must perform significant manipulation of the input data to derive correct inflecting-language output. Lacking straightforward word correspondences between the two languages, the translation model requires more complex lexical fertility and distortion relations.

- Morphologically complex languages are less studied and hence, lack the wealth of resources that would be helpful for building a better MT system. This turned out to be a major issue when translating from English to Korean, which is the main task that is done in this thesis. The Korean-English parallel corpus that was available to us when building the MT system was relatively small (around 60,000 parallel sentences) and had several sentence pairs with an inaccurate translation. This issue ended up providing a significantly negative effect to the performance of our system. In addition to monolingual or parallel textual resources, other helpful resources include corpora that has been annotated for syntactic, semantic, or other information. It also includes analytic resources that can generate supplemental levels of information on a text or language, such as sentence parsers or word analyzers. Of the previous work done on introducing morphological information to SMT, much has been built upon supervised morphological analysis used to train the model. However, these tools require human annotation and are limited in their coverage; moreover, for many languages, they are not available.

- There is no single automatic MT evaluation that can evaluate morphologically complex languages accurately. The current standard evaluation measures consider translation aspects such as the number of correct n-grams (Papineni et al., 2002). Since words in these measures are treated as unanalyzed strings, getting any single piece of inflection wrong in complex words that consist of multiple morphemes would discount the entire word and contribute to a lower evaluation score. There is also another evaluation method that uses the edit distance in words between the translation and the reference to determine the translation quality. Although some are supplemented with the ability to back off to word stems to find a match, this is available only for few languages with the necessary resources for linguistic analysis. Also, given the fact that these morphologically complex languages tend to have freer word orders, translations may be equally correct even if the word orders are different. However, such case would be penalized in the latter evaluation method.

1.3 Morpheme Segmentation for SMT

As mentioned previously, one of the biggest challenges in translating from or into morphologically complex language is that many words consist of multiple morphemes. One way to solve this problem is preprocessing the data by segmenting words from morphologically complex languages into adequate morphemes. A question arises from this last statement. What exactly is an “adequate” morpheme segmentation? Segmenting every word into the smallest unit of morphemes is not necessarily an “adequate” morpheme segmentation for SMT, since some of morphemes may not have a matching counterpart in the other language.

In English, such segmentation is obvious, since there are clear boundaries in between words. Chinese, which is also considered as an isolating language, does not incorporate any space in its sentences, but there is less ambiguity where word boundaries lie when compared to more agglutinative languages. In languages such as Hungarian, Japanese, and Korean, there is more ambiguity about where word boundaries should be. For example, consider a Korean “word” *meok-eoss-da*, which means *ate*. Unlike its English counterpart, this Korean “word” consists of three different morphemes: *eat-past-indicative*. If one uses morphological analysis as the basis for Korean segmentation, then it would segment this word into three words. This would not be desirable when translating from or into English, since English does not have counterparts for some morphemes. However, a Hungarian word *szekrényemben*, which means *in my closet*, consists of three morphemes: *closet-my-inessive*. In this case, each morpheme is a distinct word in English. Hence, in such case, we want to use morphological analysis as the basis for segmentations and segment this Hungarian word into *szekrény em ben*.

1.4 SMT for Korean

The Korean language is classified as agglutinative in its morphology, which means that most words are formed by joining morphemes together, and SOV in its syntax, which means that sentences usually appear in subject-object-verb order. The Korean language is an interesting language in that it has its own writing system, called *hangul*, and its position in the tree reflecting language families is highly debated, although some are claiming that it is a member of Altaic language, which also includes Turkic and Mongolic languages (Blench, 2008).

Apart from the challenges of morphologically complex languages in SMT, which are mentioned from the previous section, one challenge is that there are significant number of sentences that do not necessarily include a subject in them, as can be seen from Figure 1.2. This would cause the MT system to suffer from a source-target asymmetry, which would ultimately affect the translation quality.

Korean Sentence: *geureona silpaehaetda* .
Original English Translation: but he has failed .

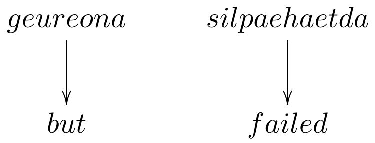

Figure 1.2: Example of a Korean sentence that does not include subject.

Being a morphologically complex language, many Korean words appear in the form of stem-suffix format. From the example *silpaehaetda*, *silpae* is the stem, which means *fail* in English, and *haetda* is the suffix, where *haet* shows that this is a past tense and *da* is an indicative morpheme. This means that there is less need for considering other types of affixes, such as prefix or infix. We used this fact when analyzing our data so that the system can focus more on dividing words into stems and suffixes when necessary.

1.4.1 Related Work

In the past, there were several different works on translating from Korean to English. Some of notable works are done by (Chung and Gildea, 2009), which is explained in detail in Chapter 3, and Korean NLP group at University of Pennsylvania [1]. In University of Pennsylvania's Korean NLP group, they used the model described in (Palmer, Rambow, and Nasr, 1998) for English/French translation to build a system that has a plug-and-play architecture that is composed of different components in parsing and generation. Their system is a hybrid system that profits from a stochastic parser that was independently trained on domain-general corpora and a hand-crafted linguistic

[1] http://www.cis.upenn.edu/ ˜xtag/koreantag

knowledge base in the form of a predicate-argument lexicon and linguistically sophisticated transfer rules. For defining transfer rules, they used the ‘lexico-structural transfer’ framework, which is based on a lexicalized predicate-argument structure. In this framework, the transfer lexicon does not simply relate words (or context-free rewrite rules) from one language to words (or context-free rewrite rules) from another language. Instead, lexemes and their relevant syntactic structures (essentially, their syntactic projection along with syntactic/semantic features) are mapped. Their data set contains a Korean-English parallel corpus in military-related topic, which consists of roughly 50,000 word tokens and 5,000 sentences. This is significantly smaller than the data set that we used for our experiments, which consists of roughly 2,000,000 words and 60,000 sentences from various news articles. Because this work is used for translating from Korean to English and the data set is domain-specific, it is not comparable to our work, which translates from English to Korean with data set that has more general topic.

When it comes to translating from English to Korean, which is what we are doing in our experiments, not much of recent work is available in English. There is one notable English paper that worked on this subject (Choi and Kim, 2007), but this is focused on translating one specific domain, patent in this case, and its translation quality was measured manually by professional translators. This makes it impossible to compare our work against theirs, since our work deals with more general domain and uses automatic evaluation for translation quality. Although there are some papers written in Korean that worked on this subject in more general sense, their data sets often are not available to public. This lack of resources and previous work contributed to difficulties in our task of translating from English to Korean.

1.5 BLEU Scores

Human evaluations of MT weigh many aspects of translation, including adequacy, fidelity, and fluency of the translation (Hovy, 1999; White and O’Connell, 1994). Although human evaluations are the most accurate ways for evaluating translation quality, for the most part, they are quite expensive (Hovy, 1999). If we need to evaluate translation output for thousands and millions of sentences, it would cost a large amount of time and money. Moreover, they can take weeks or months to finish. This is a big problem for developers of MT systems, since they need to monitor the effect of daily changes to their systems in order to weed out bad ideas from good ideas (Papineni et al., 2002).

In order to solve this problem, previous works have introduced several different automated methods for evaluating translation. One of the most popular automatic evaluations is called BLEU (Bilingual Evaluation Understudy) and it uses the number of correct n-grams (Papineni et al., 2002). This evaluation is shown to be closely correlated with human judgments of translation quality.

BLEU uses a modified form of precision to compare a candidate translation against multiple reference translations. To compute a “standard” unigram precision, one simply counts up the number of candidate translation words (unigrams) which occur in any reference translation and then divides it by the total number of words in the candidate translation. The following example, used by (Papineni et al., 2002), would get $\frac{7}{7}$ in standard unigram precision:

- Candidate: the the the the the the the.
- Reference 1: The cat is on the mat.
- Reference 2: There is a cat on the mat.

(Papineni et al., 2002) modified this method of computing precision. In order to compute the modified n-gram precision, one first counts the maximum number of times an n-gram occurs in any single reference translation. Next, one clips the total count of each candidate n-gram by its maximum reference count ($\mathit{Count}_{\mathit{clip}} = \min(\mathit{Count}, \mathit{MaxRefCount})$), adds these clipped counts up, and divides by the total (unclipped) number of candidate n-grams. Using this modified n-gram precision, we get a unigram precision of $\frac{2}{7}$ from Reference 1, which is a fairer score than $\frac{7}{7}$ from the standard unigram precision. In bigram precision, this example gets precision of 0 in modified bigram precision. This sort of modified n-gram precision scoring captures two aspects of translation: adequacy and fluency. A translation using the same words (unigrams) as in the references tends to satisfy adequacy. The longer n-gram matches account for fluency.

So far, this modified n-gram precision was done on a single sentence. Since many MT tasks involve multi-sentence test set, (Papineni et al., 2002) came up with a method for computing modified n-gram precision on blocks of text. One first computes the n-gram matches sentence by sentence. Next, one adds the clipped n-gram counts for all the candidate sentences and divide by the number of candidate n-grams in the test corpus, denoted as n-gram’ in the following equation, to compute a modified precision score, p_n, for the entire test corpus:

$$p_n = \frac{\sum_{C \in \{\mathit{Candidates}\}} \sum_{\mathit{n\text{-}gram} \in C} Count_{\mathit{clip}}(\mathit{n\text{-}gram})}{\sum_{C' \in \{\mathit{Candidates}\}} \sum_{\mathit{n\text{-}gram'} \in C'} Count(\mathit{n\text{-}gram'})}$$

In addition to this modified n-gram precision, BLEU also takes sentence length into account. Consider the following example:

- Candidate: of the
- Reference 1: It is a guide to action that ensures that the military will forever heed Party commands.
- Reference 2: It is the guiding principle which guarantees the military forces always being under the command of the Party.
- Reference 3: It is the practical guide for the army always to heed the directions of the party.

Although it is obvious that the candidate is a bad translation, it gets an inflated precision: a modified unigram precision of 2/2 ('of' and 'the') and a modified bigram precision of 1/1 ('of the'). This comes from the fact that the modified n-gram precision does not explicitly account for a recall. In order to compensate for this, BLEU includes a brevity penalty to keep scores of translations, which are composed of reference words but are shorter than the reference, from scoring artificially highly. The brevity penalty decreases the candidate's score proportionally to how much shorter it is than the reference. The brevity penalty (BP) is computed as the following:

$$BP = \begin{cases} 1 & \text{if } c > r \\ e^{(1-\frac{r}{c})} & \text{if } c \leq r \end{cases}$$

Using this brevity penalty, BLEU is formulated as:

$$BLEU = BP \cdot \exp(\sum_{n=1}^{N} w_n \log p_n)$$

and in log domain:

$$\log BLEU = \min(1 - \frac{r}{c}, 0) + \sum_{n=1}^{N} w_n \log p_n$$

where w_n are positive weights for each order n-gram, summing to one.

Since BLEU keeps track of the number of reference n-grams that appear in system output translations, in theory, the best score is 100%. In practice, however, scores are much lower. Best systems tend to achieve scores in high 30% to 40% range. For example, (Chiang, Knight, and Wang, 2009) score as high as 40.6% BLEU on a Chinese to English translation task.

In this thesis, we used BLEU for evaluating the translation quality of our system. Later in this thesis, we introduce a couple of modified versions of BLEU, which are also used for evaluating our system.

1.6 Thesis Outline

Here is how the remaining thesis will be organized. Chapter 2 talks about related previous works that are shown to improve translation quality through unsupervised morpheme segmentation and alignment methods. In Chapter 3, we describe the idea of the bilingual segmentation, which was motivated by the idea presented from (Chung and Gildea, 2009). In Chapter 4, we look into the details behind our experiments of English-to-Korean translation and present experiment results. Finally in Chapter 5, we summarize the thesis and present ideas for possible future works.

Chapter 2

Related Work

When translating between unrelated languages, disparate morphological systems can place an asymmetric conceptual burden on words, making the lexicon of one language much more coarse (Naradowsky and Toutanova, 2011). As we have seen from the last chapter, such issues can cause problems in translation between unrelated languages.

In this chapter, we will review the existing methods of unsupervised segmentation described in (Creutz and Lagus, 2005) and (Naradowsky and Toutanova, 2011). In (Creutz and Lagus, 2005), they introduce a tool called 'Morfessor', which is a monolingual unsupervised morpheme segmenter. In (Naradowsky and Toutanova, 2011), they describe an unsupervised dynamic graphical model approach for morphological segmentation and bilingual morpheme alignment for SMT. They show that their model outperforms other previous works, such as (Snyder and Barzilay, 2008), in the task of morpheme segmentation and bilingual morpheme alignment.

2.1 Monolingual Segmentation - Morfessor

Morfessor (Creutz and Lagus, 2005) uses minimum description length (MDL) and maximum a posteriori (MAP) techniques to create a model optimized for accuracy with minimal model complexity to perform segmentation without the benefit of hand-segmented training data.

Morfessor first begins with the MDL-based alrogithm, which, given an input corpus, defines a lexicon of morphs that can be concatenated to produce any word in the corpus. The MDL algorithm iterates over each input word, considering every possible split as a morph to be added to the lexicon, including the word as a whole, and selecting that split with the highest probability. This algorithm does not take where the string occurs into account and only considers the frequency of it in the text.

It models words as HMMs having only one category, which emits the context-independent morph. In this sense, it is similar to monolingual model by (Chung and Gildea, 2009). This process continues recursively until the overall probability of the split corpus converges, yielding a segmentation based upon a flat lexicon of morphs without internal substructure. As a consequence of this algorithm, frequent word forms remain unsplit, while rare word forms are over-segmented.

After the MDL algorithm, Morfessor moves to the Categories-MAP model. In this model, segmentations from MDL are reanalyzed into a recursive hierarchical morph lexicon using a greedy search algorithm. This algorithm also represents words as HMMs, but the hidden states are four latent morph categories: prefixes, stems, suffixes, and an additional "noise" category. Whether a morph is likely to function as any of these categories is determined by its meaning, which corresponds to features collected about the usage of the morph within words. The model is expressed in a MAP framework, where the likelihood of category membership follows from the usage parameters through prior probability distributions. Figure 2.1 shows hierarchical representations obtained for an English word 'straightforwardness'. Categories-MAP model makes use of word frequency information and in this case, this English word has been frequent enough in the corpus to be included in the lexicon as an entry of its own. At the same time, the inner structure of the words is retained in the lexicon, because the morphs are represented as the concatenation of other submorphs, which are also present in the lexicon. This structure is very useful in the task of morpheme segmentation, as we do not want to include non-morphemes (morphs that are tagged with 'NON') in an ideal segmentation. Hence, the finest segmentation that does not contain non-morphemes would be the output segmentation of this model. In the example from Figure 2.1, because 'for' is tagged with 'NON', the English word is expanded into 'straight+forward+ness', not 'straight+for+ward+ness'.

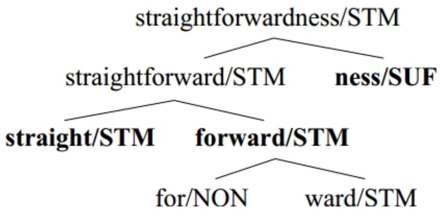

Figure 2.1: The hierarchical segmentations of 'straightforwardness'. Figure from (Creutz and Lagus, 2005).

When tested a Finnish corpus with 16 million words and 1.4 million different word forms, this algorithm achieves an F-measure of near 70%, which is better than competing unsupervised segmenters back then.

2.2 Bilingual Segmentation

2.2.1 Background

In Bulgarian, adjectives may contain markings for gender, number, and definiteness. For example, an English word ‘red’ can be written in Bulgarian in nine different forms, as can be seen from Figure 2.2. In comparison to a language which is not morphologically productive on adjectives, such as English, the alignment model must observe nine times as much data to yield a comparable statistic. Such sparsity can be quite problematic if the amount of available data plays a large role in a system’s overall performance. This creates further complications when lexical sparsity is compounded with the desire to build up alignments over increasingly larger contiguous phrases. Hence, the goal is to find the best target segmentation and alignment to source morphemes when translating from resource-rich languages, such as English, to resource-poor languages.

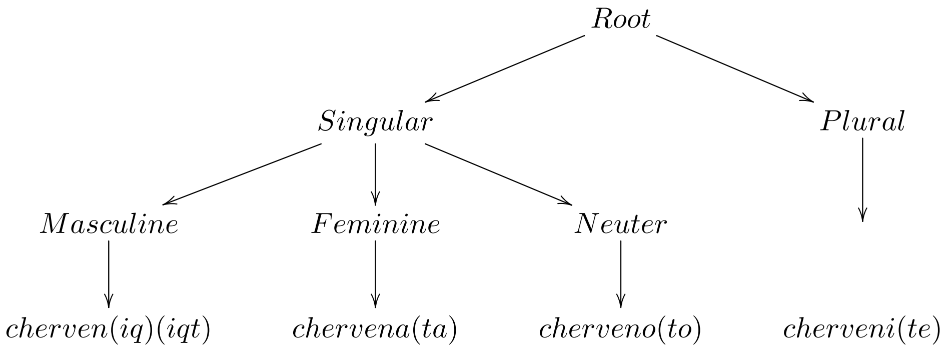

Figure 2.2: Bulgarian forms of ‘red’ (Naradowsky and Toutanova, 2011)

2.2.2 Model

In their model, the source-side input, which they assume to be English, is processed with a gold morphological segmentation, part-of-speech, and dependency tree analysis. Their model is derived from the hidden-markov model for word alignment (Vogel, Ney, and Tillmann, 1996; Och and Ney, 2000). Based on it, they define a dynamic graphical model, which lets them encode more linguistic intuition about morpheme segmentation and alignment. An example of such idea is shown in Figure 2.3.

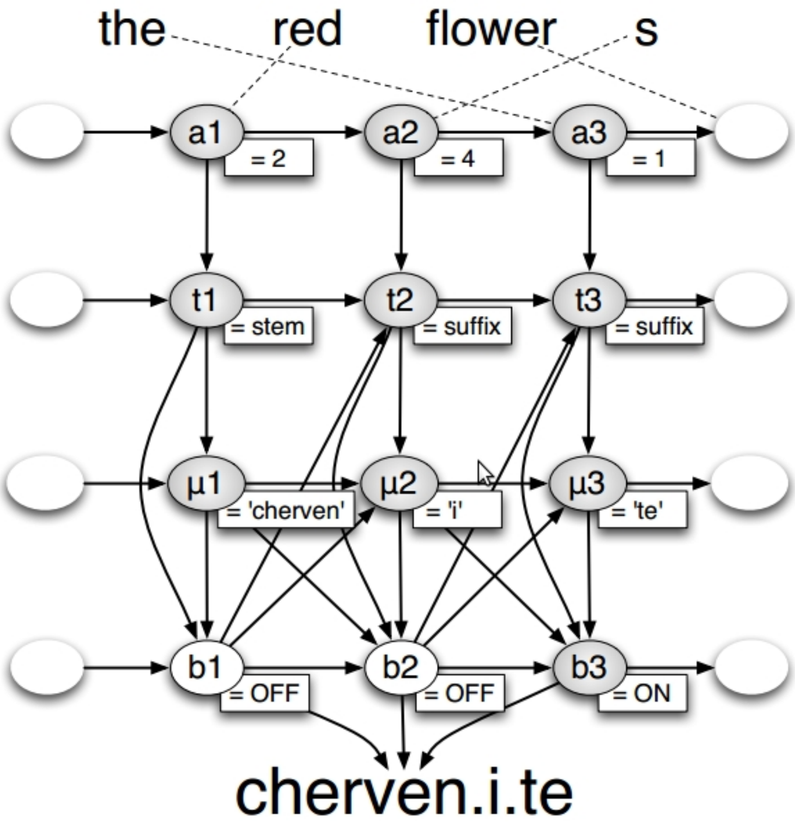

Figure 2.3: A graphical depiction of the model generating the transliteration of a Bulgarian word '*chervenite*'. Trigram dependencies and some incoming/outgoing arcs have been omitted for clarity. Figure from (Naradowsky and Toutanova, 2011).

The model is based on four different components: morpheme translation model, word boundary generation model, distortion model, and length penalty. The desired probability of target morphemes, morpheme types, alignments, and word boundaries given source is defined as the following:

$$\begin{aligned} P(\mu, \mathbf{ta}, \mathbf{b}|\mathbf{e}) &= \prod_{i=1}^{I} P_T(\mu_i|ta_i, b_{i-1}, b_{i-2}, \mu_{i-1}, \mathbf{e}) \\ &\quad \cdot P_B(b_i|\mu_i, \mu_{i-1}, ta_i, b_{i-1}, b_{i-2}, \mathbf{e}) \\ &\quad \cdot P_D(ta_i|ta_{i-1}, b_{i-1}, \mathbf{e}) \cdot LP(|\mu_i|) \end{aligned}$$

where P_T, P_B, P_D, and LP are explained in Sections 2.2.3, 2.2.4, 2.2.5, and 2.2.6.

Here is the notation that is used for describing this equation:

- $\mu_1\mu_2\ldots\mu_I$: I morphemes in the segmentation of the target sentence
- b_i: Bernoulli variable indicating whether there is a word boundary after morpheme μ_i
- c_t: non-space character in the target string
- wb_t: Bernoulli variable indicating whether there is a word boundary after the corresponding target character
- s_t: Bernoulli segmentation variable indicating whether there is a morpheme boundary after the corresponding character
- e_j: observed source language morphemes
- a_i: source morpheme aligned to μ_i
- t_i: morpheme type of μ_i (one of prefix, suffix, or stem)
- $ta_i = [a_i, t_i]$

2.2.3 Morpheme Translation Model

In the model equation, the morpheme translation probability is denoted by:

$$P_T(\mu_i|ta_i, b_{i-1}, b_{i-2}, \mu_{i-1}, \mathbf{e})$$

When multiple conditioning variables are used, they assume a special linearly interpolated back-off form of the model, similar to models routinely used in language modeling.

Suppose the morpheme translation probability is estimated as $P_T(\mu_i|e_{a_i}, t_i)$. This value is estimated in the M-step, given expected joint counts $c(\mu_i, e_{a_i}, t_i)$ and marginal counts derived from these as follows:

$$\begin{aligned}
P_T(\mu_i|e_{a_i}, t_i) &= \frac{c(\mu_i, e_{a_i}, t_i) + \alpha_2 P_2(\mu_i|t_i)}{c(e_{a_i}, t_i) + \alpha_2} \\
P_2(\mu_i|t_i) &= \frac{c(\mu_i, t_i) + \alpha_1 P_1(\mu_i)}{c(t_i) + \alpha_1} \\
P_1(\mu_i) &= \frac{c(\mu_i) + \alpha_0 P_0(\mu_i)}{c(\cdot) + \alpha_0}
\end{aligned}$$

2.2.4 Word Boundary Generation Model

In the model equation, the probability of generating word boundaries is denoted by:

$$P_B(b_i|\mu_i, \mu_{i-1}, ta_i, b_{i-1}, b_{i-2}, \mathbf{e})$$

Although the basic hidden semi-markov model ignores word boundaries, they can be useful predictors of morpheme segments. For example, they can indicate that common prefixes follow word boundaries, or that common suffixes precede them. The log-linear model of (Poon, Cherry, and Toutanova, 2009) uses word boundaries as observed left and right context features, and Morfessor (Creutz and Lagus, 2005) includes boundaries as special boundary symbols, which can inform about the morpheme state of a morpheme.

Including this distribution P_B in the model equation allows us to estimate useful information, such as the number of morphemes in a word and what morphemes are likely to have which position in a word. Through the included factored state variable ta_i, word boundaries can also inform about the likelihood of a morpheme aligned to a source word of a particular POS tag to end a word.

2.2.5 Distortion Model

In the model equation, the distortion modeling distribution is denoted by P_D. Traditional distortion models represent $P(a_i|a_{i-1}, \mathbf{e})$, the probability of an alignment given the previous alignment, to bias the model away from placing large distances between the aligned tokens of consecutively sequenced tokens. In addition to modeling a larger state space to also predict morpheme types, they extend this model by using a special log-linear model form, which allows the integration of rich morpho-syntactic context:

$$P_D = \frac{e^{\phi(a_i, a_{i-1}, \mathbf{e})}}{\sum_i e^{\phi(a_i, a_{i-1}, \mathbf{e})}}$$

Log-linear models have been previously used in unsupervised learning for local multinomial distribution like the one in (Berg-Kirkpatrick et al., 2010), and for global distributions in (Poon, Cherry, and Toutanova, 2009). The special log-linear form allows the inclusion of features targeted at learning the transitions among morpheme types and the transitions between corresponding source morphemes. Examples of such features are presented in Table 2.1.

2.2.6 Length Penalty

The last component of the model equation is the length penalty, which is denoted by $LP(|\mu_i|)$. Following (Chung and Gildea, 2009) and (Liang and Klein, 2009), they use an exponential length

Feature	**Value**
MORPH DISTANCE	1
WORD DISTANCE	1
BINNED MORPH DISTANCE	fore1
BINNED WORD DISTANCE	fore1
MORPH STATE TRANSITION	suffix-root
SAME TARGET WORD	False
POS TAG TRANSITION	DET-NN
DEP RELATION	DET $\leftarrow$ NN
NULL ALIGNMENT	False
conjunctions of above features	...

Table 2.1: Features in log-linear distortion model firing for the transition from *te*:suffix:1 to *tsvet*:root:3 in the example sentence pair in Figure 1a of (Naradowsky and Toutanova, 2011). Table from (Naradowsky and Toutanova, 2011).

penalty on morpheme length to bias the model away from the maximum likelihood under-segmentation solution:

$$LP(|\mu_i|) = \frac{1}{e^{|\mu_i|^{l_p}}}$$

where l_p is a hyper-parameter indicating the power that the morpheme length is raised to.

2.2.7 Inference

Inference is performed by EM training on the aligned sentence pairs. In the E-step, they compute expected counts of all hidden variable configurations that are relevant for the model. In the M-step, they re-estimate the model parameters with LBFGS (Limited-memory Broyden-Fletcher-Goldfarb-Shanno) method (Liu and Nocedal, 1989) for the distortion model and interpolation counts for morpheme translation and word boundary generation models.

Since the inference algorithm is quite expensive, a method for pruning out considered morpheme boundaries was introduced:

Given the target side of the corpus, derive a list of top K most frequent affixes using a simple trie-based method proposed by (Schone and Jurafsky, 2000). After determining a list of allowed prefixes and suffixes, restrict the model to allow only segmentations of the form :$((p*)r(s*))+$ where p and s belong to the allowed prefixes and suffixes and r can match any substring.

2.3 Chapter Summary

In this chapter, we reviewed previous works done by (Creutz and Lagus, 2005) and (Naradowsky and Toutanova, 2011).

In (Creutz and Lagus, 2005), Morfessor, which is a monolingual unsupervised segmenter, was introduced. It uses a combination of MDL and MAP models to effectively generate morpheme segmentations.

In (Naradowsky and Toutanova, 2011), using four different components (morpheme translation, word boundary generation, distortion,and length penalty), they introduced an unsupervised model for morpheme segmentation and alignment based on Hidden Semi-Markov Models. This new model is shown to outperform the previous works in both morpheme segmentation and alignment tasks, as can be seen from Figure 2.4 and 2.5. When tested for segmenting Hebrew and Arabic data, (Naradowsky and Toutanova, 2011) were able to obtain the highest F-score in every data set. When tested for word alignments for an Arabic data, they were able to obtain the lowest alignment error rate among other systems, such as (Chung and Gildea, 2009) and (He, 2007).

In next chapter, we introduce a method of bidirectional segmentation, which is based on the work in (Chung and Gildea, 2009).

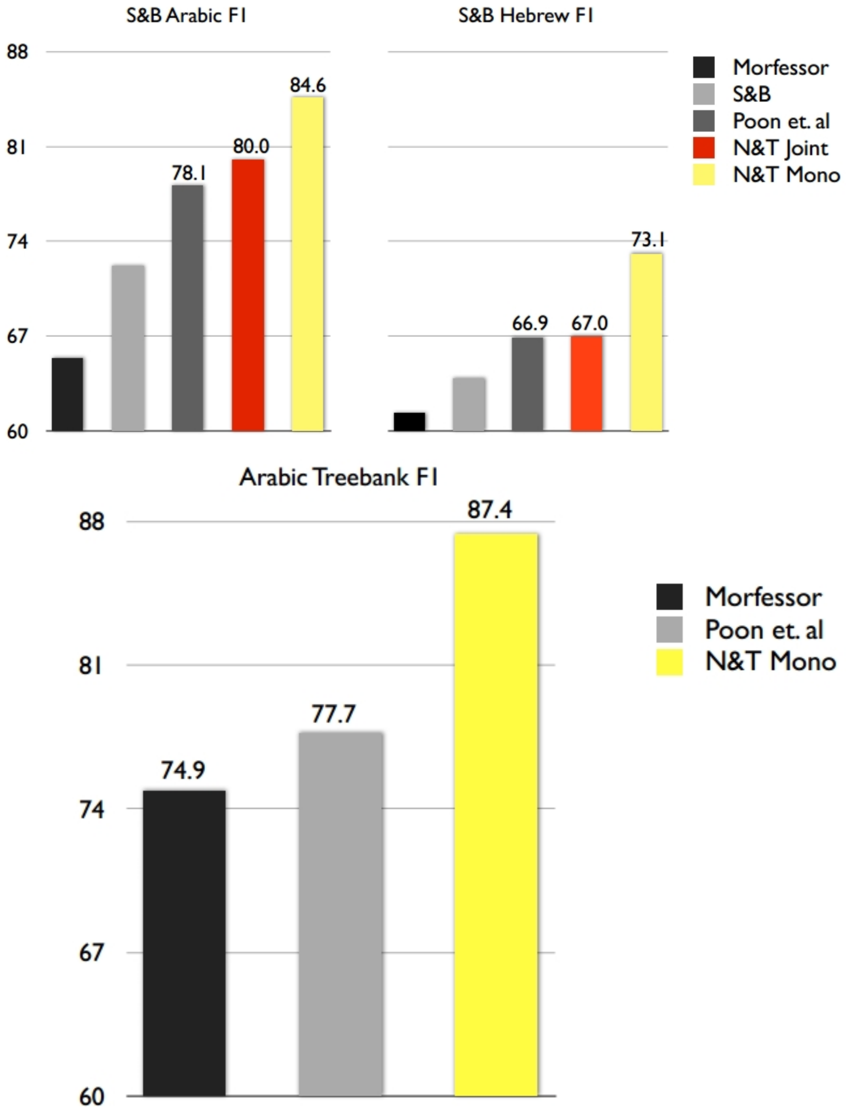

Figure 2.4: Comparison of morpheme segmentation F-scores in between Morfessor (Creutz and Lagus, 2005), S&B (Snyder and Barzilay, 2008), Poon et. al (Poon, Cherry, and Toutanova, 2009), and N&T (Naradowsky and Toutanova, 2011) for Hebrew and Arabic bible data (Snyder and Barzilay, 2008), and Arabic Tree Bank. Figures from (Naradowsky and Toutanova, 2011).

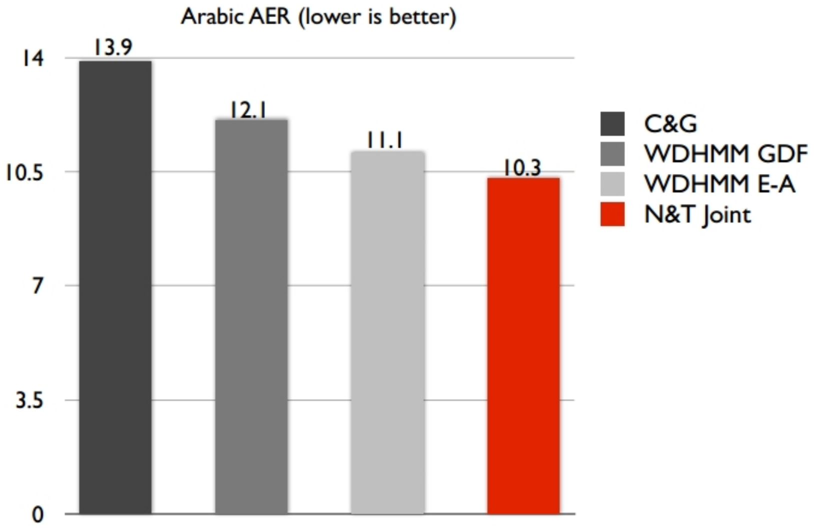

Figure 2.5: Comparison of Alignment Error Rate (AER) in between C&G (Chung and Gildea, 2009), WDHMM(He, 2007), and N&T (Naradowsky and Toutanova, 2011). Figure from (Naradowsky and Toutanova, 2011).

Chapter 3

Models

(Chung and Gildea, 2009) present two different morpheme segmentation methods that can improve Korean-English machine translation results: monolingual and bilingual models. In this thesis, these two models are used for testing on the novel task of English-Korean translation. Unlike many other MT tasks, where English is the target language, English is the source language in this experiment. We list below some of the notation we use in this paper:

- $\mathbf{c}_1^n$: Unsegmented string of Korean characters (n characters).
- e: Segmented Korean word. Similarly **e** represents a sentence formed with these segmented words.
- $\mathbf{f}_1^m$: Segmented string of English (m English words).
- **a**: Word-level alignments in between Korean and English parallel sentences. We assume that all word-level alignments are equally likely and hence $P(a) = \frac{1}{n}$. IBM Model 1 (Brown et al., 1993) was used to assume that each Korean word is generated independently from one English word.
- **s**: Segmentation where $s_i = 1$ is indicating that there is a morpheme boundary after the i-th character and $s_i = 0$ is indicating that there is no morpheme boundary after the i-th character. Hence, a string of Korean words $\mathbf{e}_1^l$ can be thought of as the result of applying the tokenization **s** to a character string **c**:

$$\mathbf{e} = \mathbf{s} \circ \mathbf{c} \quad \text{where} \quad l = \sum_{i=1}^{m} s_i$$

An example that demonstrates the above detail can be found in Figure 3.1.

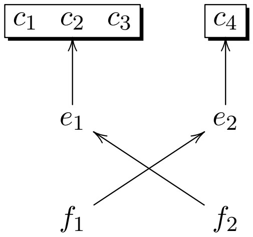

Figure 3.1: This figure shows a Korean sentence $\mathbf{e} = e_1, e_2 = \mathbf{s} \circ c_1 \dots c_4$ where $\mathbf{s} = (0, 0, 1, 1)$ and an English sentence $\mathbf{f} = f_1, f_2$. The segmentation between c_3 and c_4 means that c_1, c_2, c_3 forms the Korean word $e1$ and c_4 forms the next word e_2. These Korean words e_1 and e_2 are generated by the English words f_1 and f_2, respectively.

3.1 Monolingual Model

This model is called "monolingual" because, unlike the bilingual model, which is explained in next section, it only requires information about Korean language. The idea of monolingual model comes from learning segmentation from substring counts. Given a corpus of only source language with unknown segmentation, we want to find the optimal **s** given **c**. In particular, we want to find the **s** that gives us the highest $P(\mathbf{s}|\mathbf{c})$. According to Bayes' rule:

$$P(\mathbf{s}|\mathbf{c}) \propto P(\mathbf{c}|\mathbf{s})P(\mathbf{s})$$

Let $\mathbf{e} = \mathbf{s} \circ \mathbf{c} = e_1 \dots e_l$, where e_i is a word under some segmentation **s**. In this model, we want to find the segmentation **s** that maximizes $P(\mathbf{e})$ where $P(\mathbf{e})$ can be defined as:

$$P(\mathbf{e}) = P(e_1) \times \dots \times P(e_l)$$

When calculating $P(e_i)$, (Chung and Gildea, 2009) count every possible substring - every possible segmentation of characters - from Korean sentences. We modified this slightly to divide each word from Korean sentences into *stem* and *suffix*. From this, we count the stem and all possible substrings of the suffix. This method is employed because many Korean words follow the format of stem-suffix. Hence, this method should be modified again when tested on other languages. This modified method is explained more in detail in next chapter.

Calculating $P(e_i)$ is same as initializing $P(e|f)$ for the bilingual model, which is:

$$P(e_i) = \frac{count(e_i)}{\sum_k count(e_k)}$$

With $P(e)$ and sequence of characters **c**, we can calculate the most likely segmentation by using the Viterbi algorithm:

$$\mathbf{s}^* = \arg\max_{\mathbf{s}} P(\mathbf{e})$$

3.2 Bilingual Model

This model is called “bilingual” because it requires the information about both source and target languages. (Chung and Gildea, 2009) use IBM Model 1 (Brown et al., 1993) as the word-level alignment model, following its assumption that each foreign word is generated independently from one English word:

$$\begin{aligned} P(\mathbf{e}|\mathbf{f}) &= \sum_{\mathbf{a}} P(\mathbf{e}, \mathbf{a}|\mathbf{f}) \\ &= \sum_{\mathbf{a}} \prod_{i} P(e_i|f_{a_i}) P(\mathbf{a}) \\ &= \prod_{i} \sum_{j} P(e_i|f_j) P(a_i = j) \end{aligned}$$

In this equation, all word-level alignments **a** are equally likely, and hence, $P(a) = \frac{1}{n}$ for all positions.

This model starts by learning the probability of getting a Korean word e given an English word f using the expected maximization (EM) algorithm. Although IBM Model 1 has a simple EM update rule that can be used for computing posteriors for the alignment variable **a** and learning the lexical translation parameters $P(e|f)$, this cannot be applied directly because **e** is unknown. In fact, **e** ranges over an exponential number of possibilities depending on the hidden segmentation **s**. In order to solve this problem, we can apply dynamic programming over the sequence **s**. We compute the posterior probability of a word beginning at position i, ending at position j, and being generated by English word k:

$$P(\mathbf{s}_{i\ldots j} = (0, \ldots, 0, 1), a = k|\mathbf{f}) = \frac{\alpha(i) P(e|f_k) P(a = k) \beta(j)}{P(\mathbf{c}|\mathbf{f})}$$

where $e = c_i \ldots c_j$ is the word formed by concatenating characters i through j, and α is a variable indicating which English word position generated e. With resemblance to forward and backward probabilities in Hidden Markov Models (HMM), α and β are defined in the following way:

$$\alpha(i) = P(\mathbf{c}_1^i, s_i = 1|\mathbf{f})$$
$$\beta(j) = P(\mathbf{c}_{j+1}^m, s_j = 1|\mathbf{f})$$

α and β are computed using dynamic programming:

$$\alpha(i) = \sum_{l=1}^{L} \alpha(i-l) \sum_{a} P(a)P(c_{i-l+1}^{i}|f_a)$$
$$\beta(j) = \sum_{l=1}^{L} \sum_{a} P(a)P(c_{j+1}^{j+l}|f_a)\beta(j+l)$$

where L is the maximum character length. In this experiment, L is the length from the character c to the end of the pre-segmented word that it belongs to.

Once we have α and β, we can calculate expected counts of word pairs (c_i^j, f_k) by accumulating these posteriors over the data:

$$ec(c_{i+1}^{j}, f_k) += \frac{\alpha(i)P(a)P(c_{i+1}^{j}|f_k)\beta(j)}{\alpha(n)}$$

The M step then simply normalizes the counts:

$$P(e|f) = \frac{ec(e,f)}{\sum_e ec(e,f)}$$

One problem to note is that $P(e|f)$ needs to be initialized prior to the first iteration. Since probability of a Korean word e is independent from English words at this stage, we initialized $P(e|f)$ by approximating it with $P(e)$ in the following way (Chung and Gildea, 2009):

$$P(e|f) \approx P(e) = \frac{count(e)}{\sum_k count(e_k)}$$

We then used $P(e|f)$ which we used to compute the best alignment:

$$\begin{aligned}\mathbf{a}^* &= \arg\max_{\mathbf{a}} P(\mathbf{e}, \mathbf{a}|\mathbf{f}) \\ &= \arg\max_{\mathbf{a}} \prod_i P(e_i|f_{a_i})P(\mathbf{a})\end{aligned}$$

This best set of alignments $\mathbf{a}^*$ can be learned using the Viterbi algorithm and the segmentation $\mathbf{s}^*$ implied by $\mathbf{a}^*$ is the optimal segmentation of a Korean sentence $\mathbf{e}$.

One problem with the above equation is that we are learning $P(e|f)$ and this information is not necessarily present in the new data set that we want to translate. In fact, it is unlikely to have this information in the new data set. (Chung and Gildea, 2009) solves this problem by getting $P(e)$ from $P(e|f)$ and $P(f)$ and treating it like the monolingual model:

$$P(e) = \sum_e P(e|f)P(f)$$

3.2.1 Variational Bayes

(Chung and Gildea, 2009) mention a problem with the EM algorithm in the bilingual model. While the EM algorithm learns $P(e|f)$, it may memorize the training data and cause overfitting. In order to solve this problem, they used variational Bayes for hidden Markov model, described by (Beal, 2003) and (Johnson, 2007). Using this Bayesian extension, the emission probability of the bilingual model can be summarized as follows:

$$\theta_{\mathbf{e}}|\alpha \sim Dir(\alpha),$$
$$f_i|e_i = e \sim Multi(\theta_{\mathbf{e}}).$$

(Johnson, 2007) and (Zhang et al., 2008) show having small α helps to control overfitting. Hence (Chung and Gildea, 2009) set α as 10^{-6}.

(Chung and Gildea, 2009) apply variational Bayes to the traditional EM algorithm, which changes the M step for calculating the emission probability as the following:

$$\tilde{P}(f|e) = \frac{exp(\psi(ec(f,e) + \alpha))}{exp(\psi(\sum_e ec(f,e) + s\alpha))}$$

where ψ is the digamma function and s is the size of the vocabulary from which f is drawn. We set s to be the number of all possible tokens from training data. From this equation, we can see that setting α to a small value helps discounting the expected count with a help of the digamma function. Hence, having a lower α leads to a sparser solution.

During our experiment, we tried two different versions for EM algorithm within bilingual model: traditional EM algorithm explained from the previous section and this new EM algorithm with variational Bayes. Experiment results are presented in Table 4.3.

3.3 Length Penalty

(Chung and Gildea, 2009) added a parameter that can adjust to a segmenter's preference for longer or shorter tokens. This parameter is beneficial because we want our distribution of token length after segmentation to resemble the real distribution of token length and incorporate information on the number of tokens in the other language in a parallel corpus. It is also useful to further control overfitting. A noun attached with nominative case marker is very common in Korean and this may cause learning a noun that attached such morphemes when we just need to learn the noun itself.

(Chung and Gildea, 2009) experimented with two different length factors:

$$\begin{aligned}\phi_1(l) &= P(s)(1-P(s))^{l-1}\\ \phi_2(l) &= 2^{-l^{\lambda}}\end{aligned}$$

The first length factor, ϕ_1, is a geometric distribution, where l is the length of the token and $P(s)$ is a probability of segmentation between two characters. For the second factor, ϕ_2, λ is a parameter that is adjusted to change the number of tokens after segmentation. As can be seen from Figure 3.2, (Chung and Gildea, 2009) show that the second factor penalizes the longer tokens more heavily than the first geometric distribution. The value of $P(s)$ and λ can change to increase or decrease number of tokens after segmentation.

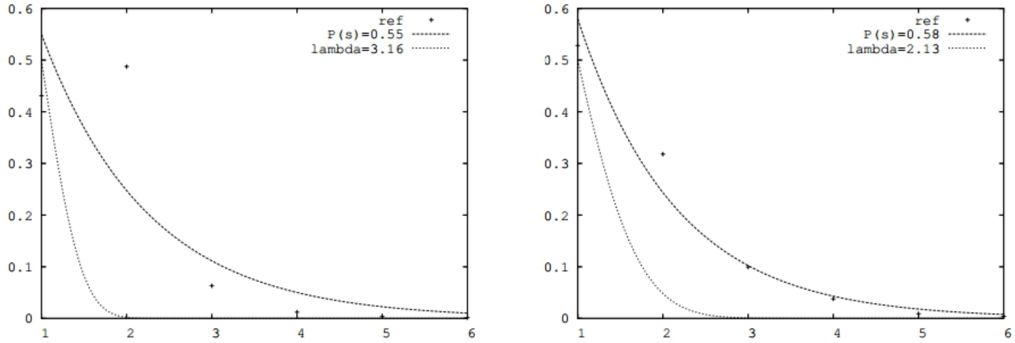

Figure 3.2: Distribution of token length for Chinese and Korean. 'ref' is the empirical distribution from supervised segmentation. Figure from (Chung and Gildea, 2009).

For monolingual model, we assume that

$$P(\mathbf{f}) \propto P(f_1)\phi(l_1) \times \ldots \times P(f_n)\phi(l_n)$$

where l_i is the length of f_i. Then we simply apply the same Viterbi algorithm to select the $f_1 \ldots f_n$ that maximizes $P(\mathbf{f})$. This set of f_i selected automatically implies the best segmentation **s** in this sentence. For $P(s)$ and λ, we pick a value that produces approximately the same number of tokens in the Korean side as the English side.

For bilingual model, each length factor is incorporated differently. For ϕ_1, the forward probability of forward-backward algorithm is modified into:

$$\alpha(i) = \sum_{l=1}^{L} \alpha(i-l)\phi_1(l) \sum_a P(a)P(c_{i-l}^{i}|f_a)$$

and the expected count of (c_i^j, f_k) is:

$$ec(c_i^j, e_k) += \frac{\alpha(i)P(a)P(c_i^j|f_k)\beta(j)\phi_1(j-i)}{\alpha(m)}$$

For ϕ_1, it is possible to fix the value of $P(s)$ or learn it in the following way:

$$P(s) = \frac{1}{m}\sum_{i}^{m}\frac{\alpha(i)\beta(i)}{\alpha(m)}$$

ϕ_2 can be incorporated into the bilingual model in the similar manner as the monolingual model. After learning $P(f)$ from the bilingual model, as shown from the previous section, we pick a λ and run the Viterbi algorithm in the same manner as the monolingual model.

In our experiment, we did not incorporate the length factor in the monolingual model. However, for bilingual model, we used ϕ_1 and fixed the value of $P(s)$ to 0.9 because it produced the best BLEU score when translating from Korean to English, as shown from the following section.

3.4 Results in Chinese-English and Korean-English Translation

(Chung and Gildea, 2009) conducted several different experiments, using monolingual and bilingual models. For Chinese-English translation, along with the monolingual and bilingual models, they used the following supervised segmenters:

- LDC segmenter [1]
- Xue's segmenter (Xue, 2003)
- Stanford segmenter (Chang, Galley, and Manning, 2008) based on Peking University's segmentation (pku) and Chinese Treebank (ctb)

For Korean-English translation, they added a rule-based morphological analyzer [2] for comparison. As one can see from Table 3.1, the unsupervised methods proposed by (Chung and Gildea, 2009) are feasible and can outperform supervised methods in some cases. Such performance is mainly driven by having better alignmnets in between English words and Korean morphemes. From this, we can observe that such methods can help with improving translation qualities when translating into morphologically-complex languages, such as translating from English to Korean.

[1] http://projects.ldc.upenn.edu/Chinese/LDC_ch.htm

[2] http://nlp.kookmin.ac.kr/HAM/eng/main-e.html

	Chinese		Korean
	BLEU	F-score	BLEU
Supervised			
Rule-based morphological analyzer			7.27
LDC segmenter	20.03	0.94	
Xue's segmenter	23.02	0.96	
Stanford segmenter (pku)	21.69	0.96	
Stanford segmenter (ctb)	22.45	1.00	
Unsupervised			
Splitting punctuation only			6.04
Maximal (Character-based MT)	20.32	0.75	
Bilingual $P(e \mid f)$ with ϕ_1 $P(s) = learned$	19.25		6.93
Bilingual $P(f)$ with ϕ_1 $P(s) = learned$	20.04	0.80	7.06
Bilingual $P(f)$ with ϕ_1 $P(s) = 0.9$	20.75	0.87	**7.46**
Bilingual $P(f)$ with ϕ_1 $P(s) = 0.7$	20.59	0.81	7.31
Bilingual $P(f)$ with ϕ_1 $P(s) = 0.5$	19.68	0.80	7.18
Bilingual $P(f)$ with ϕ_1 $P(s) = 0.3$	20.02	0.79	7.38
Bilingual $P(f)$ with ϕ_2	**22.31**	0.88	7.35
Monolingual $P(f)$ with ϕ_1	20.93	0.83	6.76
Monolingual $P(f)$ with ϕ_2	20.72	0.85	7.02

Table 3.1: BLEU score results for Chinese-English and Korean-English experiments and F-score of segmentation compared against Chinese Treebank standard. Table from (Chung and Gildea, 2009).

3.5 Bidirectional Segmentation

In the past, many different unsupervised methods for morpheme segmentation were introduced. However, their focus was on segmenting the morphologically complex side only. Even many recent works that use bilingual method of segmentation, such as (Naradowsky and Toutanova, 2011), (Snyder and Barzilay, 2008), and (Chung and Gildea, 2009), mainly focus on segmenting the morphologically complex language only. The problem with this approach is that English also has some morphological complexity and, as can be seen from rest of this section, it may be ideal to segment the English side as well to improve morpheme alignment correctness.

We now introduce a new method, named ‘bidirectional segmentation’. As the name suggests, this method involves segmenting both source and target languages. Although segmenting morphologically complex languages prior to translation helps improving the translation quality, there are cases where a segmented word may not align to an English word correctly. For example, take a Korean word ‘*kkotdeul*’. ‘*kkot*’ means ‘flower’ and ‘*deul*’ is equivalent to an English morpheme ‘s’, which is used for marking plural nouns. Ideally we would like to align ‘*kkotdeul*’ with ‘flowers’ or align ‘*kkot*’ with ‘flower’ and ‘*deul*’ with ‘s’. In the current bilingual model, because it is segmenting the morphologically complex language only, it is possible for it to segment ‘*kkotdeul*’ into two morphemes ‘*kkot*’ and ‘*deul*’, and get an incorrect alignment, where both morphemes are aligned to ‘flowers’, as can be seen from Figure 3.3. In some cases, it is even possible for ‘*deul*’ to get aligned to another word in the parallel English sentence.

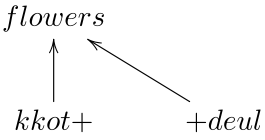

Figure 3.3: Example of incorrect alignment between an English word and morpheme-segmented Korean word. ‘+’ was used to mark the morpheme segmentation.

In order to prevent such mis-alignments from happening, we introduce a new model, where both languages are morphologically segmented. We start the model with the bilingual model, which is described from previous section. Just like the regular bilingual model, we first trained a Korean morpheme segmenter by learning $P(e|f)$ and segmented Korean sentences accordingly. Then we trained another segmenter using bilingual model again: an English segmenter, which learned $P(f|e)$ instead. We used this segmenter to segment the parallel English sentences accordingly. Using the segmented English and Korean parallel corpus, we trained a machine translator and translated our texts from English to Korean. Figure 3.4 demonstrates an overview of this bidirectional segmentation model.

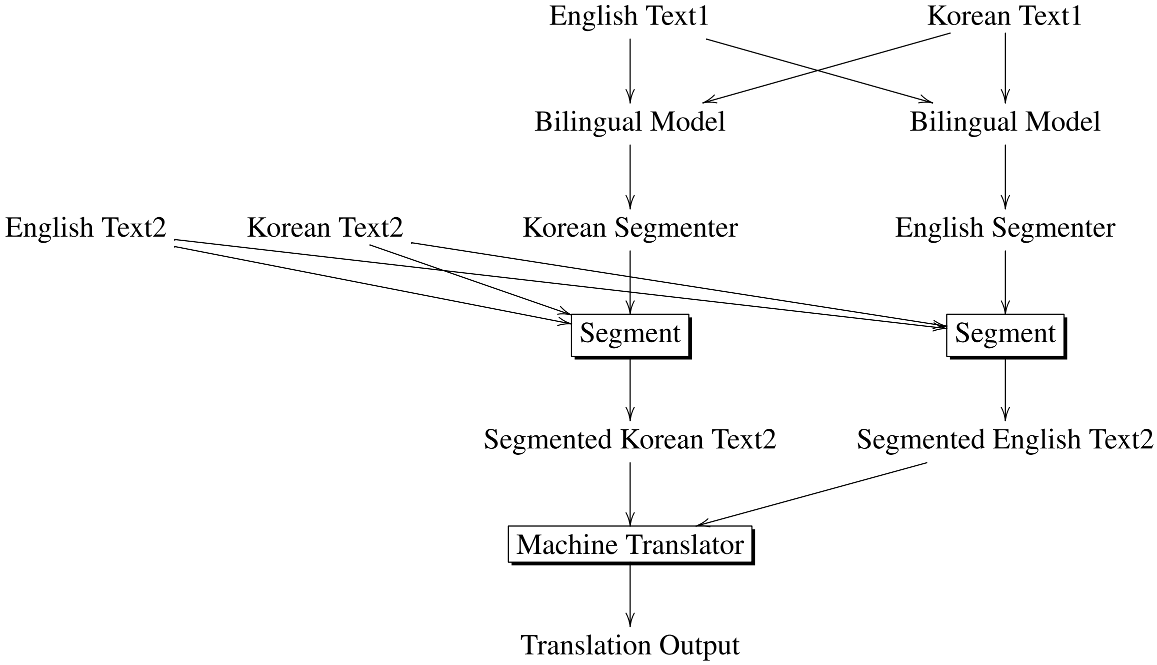

Figure 3.4: Overview of the bidirectional segmentation model.

One can argue that it is better to use the segmented Korean corpus when training the English segmenter. However, we should keep in mind that the Korean corpus that is segmented by the bilingual model does not contain perfect morpheme segmentations. In fact, some segmentations are made in obviously incorrect places and using such corpus to segment the English side may cause errors in English segmentation. Such errors may cause wrong alignments between segmented morphemes and lead to a poor translation quality.

3.6 Chapter Summary

We started this chapter by reviewing the previous work done by (Chung and Gildea, 2009). Monolingual model uses substring counts to learn segmentations, whereas bilingual model learns from alignments. From (Chung and Gildea, 2009), these models are shown to perform generally better than other unsupervised methods. We then introduced the idea of bidirectional segmentation. Observing the possible problems in alignments with single-direction segmentation, the overall idea is segmenting both source and target sides prior to translating by using the bilingual model.

In the following chapter, we will describe the experiments that were done and show the comparison of performance for each model.

Chapter 4

Experiments and Results

We have seen that the technique of bidirectional segmentation may be useful when translating morphologically complex languages. In this chapter, we describe our experiments to improve translation results through models that were mentioned from previous chapters and show that the method of bidirectional segmentation performs better than other models in some aspects.

4.1 Log Scale in Bilingual Model

One problem of dealing with probabilities in computer program is that underflows may occur frequently. Although this was not a problem for the monolingual model, since $P(e)$ can handle counts of substrings of our relatively small data without converting into log scale, this was a major issue for the bilingual model. Hence we had to work in log space when dealing with bilingual model. Forward and backward probabilities are changed into:

$$\begin{aligned}
\log \alpha(i) &= \log(\sum_{l=1}^{L} \alpha(i-l) \sum_{a} P(a) P(c_{i-l+1}^{i} | f_a)) \\
&= \log(\sum_{l=1}^{L} e^{\log \alpha(i-l)} \sum_{a} e^{\log(P(a))} e^{\log(P(c_{i-l+1}^{i}|f_a))}) \\
\log \beta(j) &= \log(\sum_{l=1}^{L} \sum_{a} P(a) P(c_{j+1}^{j+l} | f_a) \beta(j+l)) \\
&= \log(\sum_{l=1}^{L} \sum_{a} e^{\log(P(a))} e^{\log(P(c_{j+1}^{j+l}|f_a))} e^{\log(\beta(j+l))})
\end{aligned}$$

Similarly, the E step of the EM algorithm becomes:

$$\begin{aligned}\log ec(c_{i+1}^{j}, f_k) &= \log(ec(c_{i+1}^{j}, f_k) + \frac{\alpha(i)P(a)P(c_{i+1}^{j}|f_k)\beta(j)}{\alpha(n)}) \\ &= \log(e^{\log(ec(c_{i+1}^{j}, f_k))} + e^{\log(\frac{\alpha(i)P(a)P(c_{i+1}^{j}|f_k)\beta(j)}{\alpha(n)})}) \\ &= \log(e^{\log(ec(c_{i+1}^{j}, f_k))} + e^{\log\alpha(i)+\log P(a)+\log P(c_{i+1}^{j}|f_k)+\log\beta(j)-\log\alpha(n)})\end{aligned}$$

and the M step becomes:

$$\begin{aligned}\log P(e|f) &= \log(\frac{ec(e,f)}{\sum_e ec(e,f)}) \\ &= \log ec(e,f) - \log(\sum_e ec(e,f)) \\ &= \log ec(e,f) - \log(\sum_e e^{\log ec(e,f)})\end{aligned}$$

4.2 Data

We used two different sets of data for this experiment. The first set is the KAIST corpus[1], which was created by Korea Advanced Institute of Science and Technology. This data set includes about 590,000 words and 60,000 sentences on the English side. Although this data does not have as many words as the second data set, which will be explained shortly, and contains many similar-looking sentences that share many words, it was sufficient for observing various suffixes. Hence, this parallel corpus was used for training the bilingual model for segmenting.

The second set of data, referred as 'Rochester data' throughout the rest of this thesis, is the same Korean-English parallel corpus as what (Chung and Gildea, 2009) used. This data is collected from news websites and sentence-aligned using two different tools described by (Moore, 2002) and (Melamed, 1999). The data set includes:

- About 2,000,000 words and 60,000 sentences on the English side, where most of them are used as training data.
- 2,200 sentence pairs randomly sampled from the parallel corpus, where half are used as test set and the other half are used as tuning set.

This data set was used for training the monolingual model, was segmented by both bidirectional and monolingual models, and was tested for MT results.

[1] http://swrc.kaist.ac.kr

4.3 Experimental Setup

4.3.1 Moses

We used the Moses machine translation system (Koehn et al., 2007) for testing our models on MT task. This is the same system as the one that was used by (Chung and Gildea, 2009), which conducted similar experiments from this thesis. Default parameters were used throughout the translation process: hypothesis stack size 100, distortion limit 6, phrase translations limit 20, and maximum phrase length 20. For the language model, we used SRILM 5-gram language models (Stolcke, 2002) for all factors.

4.3.2 Stem and Suffix

For both monolingual and bilingual models, not only the pre-segmented words in our training data, but also words inside them had to be considered and added to the generated dictionary of Korean words. Because most of segmentation tasks in Korean language involve segmenting out correct suffixes, we created three different dictionaries based on the maximum length of suffix for each pre-segmented word: 3, 5, and 10 characters long. Based on this maximum length of suffix, each word is divided into a stem and the remaining suffix. When training the segmenter, the entire word, stem and suffix of the word, and all possible substrings of the suffix are counted. With larger maximum suffix length, the generated dictionary consisted of more words, hence providing some probability to more words. These different dictionaries with different counts would affect the value of $P(e)$ and change our results. Maximum suffix length of 3, 5, and 10 were used for monolingual model and 3 was used for bilingual model when segmenting Korean sentences. This particular length of 3 was selected for bilingual model, because most Korean suffixes are 3 or less characters long.

When segmenting English sentences for bidirectional segmentation model, maximum suffix length of 5 was used. Once again, this particular length was chosen for segmenting English, because most English suffixes are 5 or less letters long.

Figure 4.1 demonstrates how a suffix can be broken down using this method.

4.3.3 Data Set

As mentioned from the previous section, the KAIST corpus was used for training the segmenter that uses bilingual model. This was done to prevent the bilingual segmenter from preferring not to segment the data it has seen from training process. From previous chapter, we mentioned that the

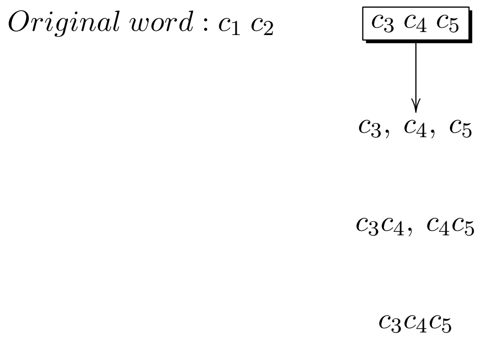

Figure 4.1: Suffix break-down of a word '$c_1c_2c_3c_4c_5$' with the maximum suffix length specified as 3.

posterior probability of a word beginning at position i, ending at position j, and being generated by Korean word k is defined as the following:

$$P(s_{i\ldots j} = (1, 0, \ldots, 0, 1), a = k|\mathbf{e}) = \frac{\alpha(i)P(f|e_k)P(a = k)\beta(j)}{P(\mathbf{c}|\mathbf{e})}$$

where α and β are defined as:

$$\alpha(i) = P(c_1^i, s_i = 1|\mathbf{e})$$
$$\beta(j) = P(c_{j+1}^m, s_j = 1|\mathbf{e})$$

In other words, $\alpha(i)$ and $\beta(j)$ refer to the forward and backward probability of having a segmentation at position i and j, respectively. During the training process for bilingual model, the value of α and β are set to 1 for the 'whole' words in training data, since the pre-existing segmentation means that there must be a segmentation. This ultimately causes the probabilities that involve these whole words to be significantly large and the segmenter will prefer to leave these words without segmenting. Hence, the data set that is different from the one to be segmented had to be used for training bilingual model.

For baseline method, we trained, tuned, and tested Moses with raw Rochester data. Hence, only segmentations in this method are pre-existing spaces in sentences.

4.3.4 Evaluation Methods

As mentioned from Chapter 1, one of the biggest issues with the existing automatic translation quality evaluation methods is that there is no single automatic MT evaluation that can evaluate morphologically complex languages accurately. Although we are using BLEU, which considers the number of correct n-grams, since words in these measures are treated as unanalyzed strings, getting any single piece of inflection wrong in complex words that consist of multiple morphemes would discount the entire word and contribute to a lower evaluation score.

For each translated test data set, except for the translation from the baseline method, we computed three different BLEU scores:

- "Regular" word-based BLEU: morpheme segmentations in the translation output, marked by '+', are merged and BLEU scores are computed against raw reference data.
- m-BLEU (morpheme-level BLEU): morpheme segmentations are kept in the translation output and BLEU scores are computed against reference data that has been segmented with the same segmenter used. (Luong, Nakov, and Kan, 2010)
- c-BLEU (character-level BLEU): 4-gram character-level BLEU scores are computed, since the average length of Korean words in Rochester data is 4 characters long. (Denoual and Lepage, 2005)

In c-BLEU, each sentence received maximal segmentation (segment every character) and hence, each character was treated like a word in the regular BLEU. As mentioned, we were able to observe that the average length of each word in Rochester data is around 4 characters long, so we used 4-gram c-BLEU. Unlike Chinese, where each character has a meaning of its own, one character in Korean is unlikely to have any meaning. So it is important to make sure that scores that we are getting from c-BLEU are coming from word-like grams. Using morpheme and character-level BLEU with the regular word-level BLEU allows us to compare how each model is doing in terms of not only words, but also morphemes and characters.

4.3.5 Merging

When merging morpheme segmentations to compute "Regular" BLEU, there were different methods for doing this. Merging segmentations that are marked by '+ +' was trivial, since we had to merge them normally. However, problem comes from translation outputs that contain only one '+' sign in between a space like the following: '*word1 +word2*'. In such case, there are two different methods for merging them. The first method is just removing the plus sign only and keeping the space in between *word1* and *word2*, which would turn the above phrase into '*word1 word2*'. The second

Original Text	*i+ +deuleun aneul 50+ +0manwon +bootuh .*
Method 1	*ideuleun aneul 500manwon bootuh .*
Method 2	*ideuleun aneul 500manwonbootuh .*

Table 4.1: Comparison of different merge methods.

method is removing the space as well as the plus sign, which would turn the above phrase into the following: '*word1word2*'. Table 4.1 lists an example that compares these two merge methods.

Because our MT system was trained with '+' as the indicator for morpheme separator, words from translation outputs that contain '+' are morphemes and should be used as parts of whole Korean words. Hence, we decided to use the second merge method when merging spaces with only one '+' sign.

4.4 Results

Table 4.2 lists sample segmented sentences from Rochester Korean-English data, which are eventually used as part of our machine translation task. As can be seen from Table 4.3, for regular word-based BLEU score, the baseline segmentation scored the lowest and our bidirectional model scored the highest. Although they outperformed baseline method, both monolingual and single-direction bilingual models scored lower than the bidirectional model. For m-BLEU scores, it may appear that the monolingual model with maximum suffix length of 3 scored the highest. However, these scores are not directly comparable, since each method has its own reference data with different segmentations. For c-BLEU scores, the monolingual model with maximum suffix length of 3 scored the highest. Although our bidirectional models surpassed the baseline method in c-BLEU score, we can observe that their performance is not as effective as other segmentation methods.

Although these BLEU scores are low when compared to other MT tasks, including (Chung and Gildea, 2009), readers should keep in mind that we did English-Korean MT instead of Korean-English. Korean, being a morphologically complex language, has a difficulty that is similar to or harder than other morphologically complex languages, such as Turkish or Finnish, when it is used as a target language in MT task. Turkish and Finnish, which traditionally use 'regular' BLEU scores for measuring MT quality, possess similar difficulties to Korean when traslating from English and tend to have relatively low BLEU scores (Oflazer and El-Kahlout, 2007; Koehn, 2005). However the point of this experiment was to show that bidirectional segmentation helps translating into

English	there is no way that software can upgrade itself just by changing the hardware .
Korean	*hadeuweuhreul bakundago sopeuteuweuhga jeojeolro ubgraeideudwel ri upda.*
Monolingual (Kor)	*hadeuwe+ +uhreul bakun+ +dago sopeuteuwe+ +uhga jeo+ +jeolro ubgeuraei+ +deudwel ri upda.*
With Variational Bayes	
Bilingual (Eng)	there is n+ +o way that software can up+ +grade itself just by changing the hardware .
Bilingual (Kor)	*hadeuweuh+ +reul bakunda+ +go sopeuteuweuh+ +ga jeojeolro ubgeuraeideudwel ri upda.*
Without Variational Bayes	
Bilingual (Eng)	there is no way that software can up+ +grade itself just by changing the hardware .
Bilingual (Kor)	*hadeuweuh+ +reul bakunda+ +go sopeuteuweuh+ +ga jeojeolro ubgeuraeideudwel ri upda.*

Table 4.2: Sample segmented sentences from Rochester Korean-English data. Spaces surrounded by '+' represent newly added segmentations. For monolingual model, we used the one with maximum suffix length of 10. The original data is in UTF-8 but is romanized here for convenience.

Method	**BLEU**	**m-BLEU**	**c-BLEU**
Baseline	3.13	N/A	20.53
Monolingual (3)	3.25	5.08	**24.18**
Monolingual (5)	3.29	4.83	23.90
Monolingual (10)	3.20	4.68	22.45
Single-Direction Bilingual (*with variational Bayes*)	3.29	4.68	21.97
Single-Direction Bilingual (*without variational Bayes*)	3.36	4.32	22.26
Bidirectional (*with variational Bayes*)	**3.46**	4.57	20.66
Bidirectional (*without variational Bayes*)	3.22	4.14	20.87

Table 4.3: BLEU score results for different methods in English-Korean machine translation. Numbers in brackets represent maximum length of suffix used when training the segmenter.

morphologically-complex languages and this turned out to be true as shown from its performance against other methods.

4.5 Chapter Summary

In this chapter, we provided a detailed explanation about overall experiments, including data sets, experiment and evaluation methods, usage of variational Bayes, and results. We started with modifying bilingual model to use a log scale, which was done to prevent underflow values from occurring. When training the segmenter, we used a different data set from the one that was segmented and translated. As explained from this chapter, this was done to improve the overall performance of the segmenter, as the segmenter prefers not to segment the words that it has seen from training. Once the data sets are segmented and translated with Moses, translation outputs are evaluated with three different BLEU scores: regular word-level BLEU, morpheme-level BLEU, and character-level BLEU. When getting word-level BLEU scores, it is essential to merge morpheme segmentations in translation outputs accurately. Although the experiment results are not as good as we had hoped for, we are able to observe that the bidirectional segmentation model gets the highest score in the word-level BLEU.

In next chapter, we will sum up this thesis by discussing possible problems with our experiments and suggest possible future works that can follow this thesis.

Chapter 5

Conclusion

One of the biggest challenges in translating into a morphologically-complex language is the lack of evaluation methods. Although we used BLEU and some of its modified forms to evaluate our translation quality, there are several synonyms in many Korean words and morphemes that are attached to Korean words are quite flexible with changes. Once an improved automatic evaluation algorithm comes out, we are certain that translating into Korean or other morphologically complex languages will look better in terms of scores.

Another problem with languages like Korean is that the amount of available resources is quite poor. The Rochester parallel corpus was quite small in size when compared to data sets used in other translation tasks. Because we are using statistical models, it obviously is better when trained on more data. Along with the small data size, our data also had some errors in reference translation. These reference translation errors contributed to low overall evaluation scores. Here is an example where the reference translation was critically mismatched:

- English: perc polled businesspeople in 13 asian countries .
- Korean: *hongkongeui gukgasinyongpyunggagigwanin jungchigyungjehwihumjamungongsa (perc) neun ahsiah jiyuk 13gaegook giubindeuleul daesangeuro gakgookeui gwanryojeh siltaereul josa bulsukhan gyulgwa hongkonge 3 . 29jumeuro gajang joeun sungjukeul gudwotdago 26il balkhyutda .*

Even those who do not understand Korean can see that these two sentences do not mean the same. Because it was impossible to go through every sentence in the data set manually and pick out such parallel sentences, a significant number of them exist in our data set and this contributed to the low evaluation scores. We believe that a large enough error-free data set would change our scores for each model significantly.

Despite such disadvantages, we were able to observe that there is a case where the translation quality score has improved with the bidirectional segmentation model. In particular, the bidirectional segmentation model received the highest BLEU score in word-level, with an improvement from a baseline score of 3.13 to 3.46 for our proposed bidirectional segmentation model.

5.1 Future Works

Here are some suggestions for possible future works that can be done after this thesis:

- We first would like to conduct the same experiment with different settings of length penalty for bilingual model. Since we only used ϕ_1 with $P(s) = 0.9$ in bilingual model for our experiments, it would be interesting to observe how the models behave when provided with different length penalties, such as a different value of $P(s)$ or ϕ_2.

- We also would like to test our algorithm in other languages that need segmentation, such as Chinese and Finnish. Because the task of segmentation is quite similar in other languages, we believe that simple modification in our program can make it usable on those languages as well.

- Another aspect that we can spend some time is finding the new method for length penalty. As mentioned from previous chapter, one of the problems in current bidirectional segmentation model is that the bilingual model, which our bidirectional model is based on, requires a separate data set to train the segmenter. If we can come up with a new length penalty model that can force the segmenter to segment words that were used in training, then there would be no need for a separate data set for training the segmenter. In fact, we believe that this would lead the segmenter to behave more correctly.

- In this thesis, we mentioned several other unsupervised methods that are used for segmenting morphologically complex languages. Along with Morfessor (Creutz and Lagus, 2005) and the works done in (Naradowsky and Toutanova, 2011), which we explained in chapter 2, there are many more different models, such as (Snyder and Barzilay, 2008). It would be interesting to try these different models on Korean and observe how it affects the English-Korean translation.

References

Beal, Matthew J. 2003. *Variational Algorithms for Approximate Bayesian Inference*. Ph.D. thesis, Gatsby Computational Neuroscience Unit, University College London.

Berg-Kirkpatrick, Taylor, Alexandre Bouchard-Côté, John DeNero, and Dan Klein. 2010. Painless unsupervised learning with features. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, HLT '10, pages 582–590, Stroudsburg, PA, USA. Association for Computational Linguistics.

Blench, Roger. 2008. Stratification in the peopling of China: how far does the linguistic evidence match genetics and archaeology? In *Human migrations in continental East Asia and Taiwan: genetic, linguistic and archaeological evidence*.

Brown, Peter F., Vincent J.Della Pietra, Stephen A. Della Pietra, and Robert. L. Mercer. 1993. The mathematics of statistical machine translation: Parameter estimation. *Computational Linguistics*, 19:263–311.

Chang, Pi-Chuan, Michel Galley, and Christopher D. Manning. 2008. Optimizing Chinese word segmentation for machine translation performance. In *Proceedings of the Third Workshop on Statistical Machine Translation*, StatMT '08, pages 224–232, Stroudsburg, PA, USA. Association for Computational Linguistics.

Chiang, David, Kevin Knight, and Wei Wang. 2009. 11,001 new features for statistical machine translation. In *In North American Chapter of the Association for Computational Linguistics - Human Language Technologies (NAACL-HLT)*.

Choi, Sung-Kwon and Young-Gil Kim. 2007. Customizing an English-Korean machine translation system for patent translation. In *Proceedings of the 21st Pacific Asia Conference on Language, Information and Computation*.

Chung, Tagyoung and Daniel Gildea. 2009. Unsupervised tokenization for machine translation. In *Conference on Empirical Methods in Natural Language Processing*, Singapore.

Clifton, Ann. 2010. Unsupervised morphological segmentation for statistical machine translation. Master's thesis, School of Computing Science, Simon Fraser University.

Creutz, Mathias and Krista Lagus. 2005. Inducing the morphological lexicon of a natural language from unannotated text. In *Proceedings of the International and Interdisciplinary Conference on Adaptive Knowledge Representation and Reasoning (AKRR05)*, pages 106–113.

Denoual, Etienne and Yves Lepage. 2005. BLEU in characters: towards automatic MT evaluation in languages without word delimiters. In *Proceedings of 2nd IJCNLP*, pages 81–86.

He, Xiaodong. 2007. Using word dependent transition models in HMM based word alignment for statistical machine translation. In *Proceedings of the Second Workshop on Statistical Machine Translation*, StatMT '07, pages 80–87, Stroudsburg, PA, USA. Association for Computational Linguistics.

Hovy, E. H. 1999. Toward finely differentiated evaluation metrics for machine translation. In *Proceedings of the Eagles Workshop on Standards and Evaluation*, Pisa, Italy.

Johnson, Mark. 2007. Why doesn't EM find good HMM POS-taggers? In *EMNLP-CoNLL*, pages 296–305.

Koehn, Philipp. 2005. Europarl: A Parallel Corpus for Statistical Machine Translation. In *Conference Proceedings: the tenth Machine Translation Summit*, pages 79–86, Phuket, Thailand. AAMT, AAMT.

Koehn, Philipp, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondrej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of ACL, Demonstration Session*, pages 177–180.

Liang, Percy and Dan Klein. 2009. Online EM for unsupervised models. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, NAACL '09, pages 611–619, Stroudsburg, PA, USA. Association for Computational Linguistics.

Liu, D. C. and J. Nocedal. 1989. On the limited memory BFGS method for large scale optimization. *Math. Program.*, 45(3):503–528, December.

Luong, Minh-Thang, Preslav Nakov, and Min-Yen Kan. 2010. A hybrid morpheme-word representation for machine translation of morphologically rich languages. In *Proceedings of the*

2010 Conference on Empirical Methods in Natural Language Processing, EMNLP '10, pages 148–157, Stroudsburg, PA, USA. Association for Computational Linguistics.

Melamed, I. Dan. 1999. Bitext maps and alignment via pattern recognition. *Computational Linguistics*, 25:107–130.

Moore, Robert C. 2002. Fast and accurate sentence alignment of bilingual corpora. In *AMTA 02: Proceedings of the 5th Conference of the Association for Machine Translation in the Americas on Machine Translation: From Research to Real Users*, pages 135–144. Springer-Verlag.

Naradowsky, Jason and Kristina Toutanova. 2011. Unsupervised bilingual morpheme segmentation and alignment with context-rich hidden semi-markov models. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 895–904, Portland, Oregon, USA, June. Association for Computational Linguistics.

Och, Franz Josef and Hermann Ney. 2000. Improved statistical alignment models. In *Proceedings of the 38th Annual Meeting on Association for Computational Linguistics*, ACL '00, pages 440–447, Stroudsburg, PA, USA. Association for Computational Linguistics.

Oflazer, Kemal and Ilknur Durgar El-Kahlout. 2007. Exploring different representational units in English-to-Turkish statistical machine translation. In *Proceedings of the Second Workshop on Statistical Machine Translation*, StatMT '07, pages 25–32, Stroudsburg, PA, USA. Association for Computational Linguistics.

Palmer, Martha, Owen Rambow, and Alexis Nasr. 1998. Rapid prototyping of domain-specific machine translation systems. In David Farwell, Laurie Gerber, and Eduard Hovy, editors, *Machine Translation and the Information Soup*, volume 1529 of *Lecture Notes in Computer Science*. Springer Berlin / Heidelberg, pages 95–102.

Papineni, Kishore, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ACL '02, pages 311–318, Stroudsburg, PA, USA. Association for Computational Linguistics.

Poon, Hoifung, Colin Cherry, and Kristina Toutanova. 2009. Unsupervised morphological segmentation with log-linear models. In *Proceedings of Human Language Technologies: The 2009*

Annual Conference of the North American Chapter of the Association for Computational Linguistics, NAACL '09, pages 209–217, Stroudsburg, PA, USA. Association for Computational Linguistics.

Schone, Patrick and Daniel Jurafsky. 2000. Knowledge-free induction of morphology using latent semantic analysis. In *Proceedings of the 2nd workshop on Learning language in logic and the 4th conference on Computational natural language learning - Volume 7*, ConLL '00, pages 67–72, Stroudsburg, PA, USA. Association for Computational Linguistics.

Snyder, Benjamin and Regina Barzilay. 2008. Unsupervised multilingual learning for morphological segmentation. In *Proceedings of ACL-08: HLT*, pages 737–745, Columbus, Ohio, June. Association for Computational Linguistics.

Stolcke, Andreas. 2002. SRILM - an extensible language modeling toolkit. In *7th International Conference on Spoken Language Processing*, pages 901–904.

Vogel, Stephan, Hermann Ney, and Christoph Tillmann. 1996. HMM-based word alignment in statistical translation. In *Proceedings of the 16th conference on Computational linguistics - Volume 2*, COLING '96, pages 836–841, Stroudsburg, PA, USA. Association for Computational Linguistics.

White, John S. and T. O'Connell. 1994. The ARPA MT evaluation methodologies: Evolution, lessons, and further approaches. In *Proceedings of the 1994 Conference of the Association for Machine Translation in the Americas*, pages 193–205, Columbia, Maryland.

Xue, Nianwen. 2003. Chinese word segmentation as character tagging. *International Journal of Computational Linguistics and Chinese Language Processing*, 8:29–48.

Zhang, Hao, Chris Quirk, Robert C. Moore, and Daniel Gildea. 2008. Bayesian learning of non-compositional phrases with synchronous parsing. In *Proceedings of ACL-08: HLT*, pages 97–105, Columbus, Ohio, June. Association for Computational Linguistics.