Chunk alignment for Corpus-Based Machine Translation

Jae Dong Kim

CMU-LTI-11-002

September 29, 2010

Language Technologies Institute
School of Computer Science
Carnegie Mellon University
5000 Forbes Ave., Pittsburgh, PA 15213
www.lti.cs.cmu.edu

Thesis Committee:
Jaime Carbonell (Chair)
Ralf Brown (Co-chair)
Stephan Vogel
Andy Way, Dublin City University

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy
in Language and Information Technologies

Keywords: Symmetric Probabilistic Alignment, Example-Based Machine Translation, Statistical Machine Translation, Phrase-Based Statistical Machine Translation, Chunk-Based Machine Translation

To my parents with love and gratitude,

Abstract

Since sub-sentential alignment is critically important to the translation quality of an Example-Based Machine Translation (EBMT) system, which operates by finding and combining phrase-level matches against the training examples, we developed a new alignment algorithm for the purpose of improving the EBMT system's performance. This new Symmetric Probabilistic Alignment (SPA) algorithm treats the source and target languages in a symmetric fashion.

We describe our basic algorithm and its primary extensions that enable use of surrounding context, and of positional preference information, compare its alignment accuracy with IBM Model 4, and report on experiments in which either IBM Model 4 or SPA alignments are substituted for the aligner currently built into the EBMT system. Both Model 4 and SPA are significantly better than the internal aligner.

Then we extend SPA to exploit external alignment information from Moses and to output non-contiguous target phrases. We also alter SPA so that the weights for its feature scores are tuned using minimum error rate training. Our experiments show that exploiting external alignment information and non-contiguous alignment are helpful for SPA in the EBMT system.

Even with these improvements, however, SPA still could not properly deal with systematic translation for insertion or deletion words between two distant languages. Therefore, we attempt to alleviate this problem by using syntactic chunks as translation units. To do so, we developed a new chunk alignment algorithm that exploits word alignment information to align chunks. Then we integrated a chunk-based translation component based on the chunk alignment into the EBMT system that uses SPA for phrasal alignment. We show that the chunk alignment performs significantly better than the baseline system that aligns two chunks if any word pair of the two chunks has word alignment link. We also demonstrate that the system with chunk-based translation is significantly better than the baseline EBMT system with SPA in translation quality.

Acknowledgments

Looking back, I would like to acknowledge the help of the generous people that made my dissertation possible.

It is difficult to overstate my gratitude to my academic advisors Jaime Carbonell and Ralf Brown. They taught me with deep and broad knowledge and advised me with enthusiasm. It was my honor to be the student of such great scholars.

I am also grateful to the other committee members. I was fortunate to have the input of respected scholars, Stephan Vogel and Andy Way, whose valuable advice made my dissertation more concrete.

I thank the people who encouraged me to go back to school. It was not an easy decision to go back to study after having worked for 6 years in the industry. However, John Kang helped me to study at LTI as a friend and advisor, and Sungwoo Park gave me advice as a friend and senior graduate student.

I am indebted to my student colleagues. Since the beginning of my graduate program, Sanjika Hewavitharana has been my best friend and given me sincere advice as both a friend and a fellow student. In the EBMT group, Aaron Phillips and Rashmi Gangadharaiah also helped me as fellow students.

As a foreign student, it was a challenge to write a thesis in sound English. I wish to thank Elizabeth-Anne Kim for reviewing my English. She was not only a good reviewer but also a good engineer, who understood my thesis and gave me advice. I was so lucky that she reviewed my dissertation.

Lastly, most importantly, I thank my family and would like to share this honor. My parents, brothers, and sisters have constantly been my greatest supporters. My wife Sunhee (Cathy) encouraged me to study again and has been an immense support and source of advice. My daughter Soo-hyun (Janise) and my son Gwang-hyun (Alex) have been my wellspring of joy and energy.

Abbreviations

ACL	Association for Computational Linguistics
AER	Alignment Error Rate
ARTFL	American and French Research on the Treasury of the French Language
BLEU	BiLingual Evaluation Understudy
CLLM	Chunk Label Language Model
CM	Chunk Mapping
CMU	Carnegie Mellon University
EBMT	Example-Based Machine Translation
EM	Expectation Maximization
FBIS	Foreign Broadcast Information Service
HMM	Hidden Markov Model
IDF	Inverse Document Frequency
ITG	Inversion Transduction Grammar
LM	Language Model
METEOR	Metric for Evaluation of Translation with Explicit ORdering
MI	Mutual Information
NIST	National Institute of Standards and Technology
NLP	Natural Language Processing
OOV	Out Of Vocabulary
PBSMT	Phrase-Based Statistical Machine Translation
POS	Part-Of-Speech
RBMT	Rule-Based Machine Translation
SLR	Sentence Level Reordering
SMT	Statistical Machine Translation
SOV	Subject-Object-Verb
SPA	Symmetric Probabilistic Alignment
SVO	Subject-Verb-Object
WMT	Workshop on Statistical Machine Translation

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Improvements we achieved

In this thesis work, we achieved substantial improvements according to automated evaluation metrics for three different language pairs in the CMU Example-Based Machine Translation (EBMT) system as shown in Table 1.1. To achieve the improvements, we investigated a new phrasal alignment algorithm and a different translation unit and introduced Statistical Machine Translation (SMT) techniques in the EBMT system. The improvements were statistically significant and consistent through two different widely used Machine Translation (MT) metrics BLEU and METEOR.

	BLEU	METEOR
Korean-English	11.16 %	7.02 %
Chinese-English	27.05 %	10.76 %
French-English	5.38 %	2.26 %

Table 1.1: The improvements we achieved

1.2 A brief history of Machine Translation

Since Andrew Booth and Warren Weaver's first attempt to use newly invented computers for machine translation appeared in 1946 and 1947, many machine translation approaches have been developed (Hutchins, 2007).

In the early days, researchers studied two main kinds of approaches. The first, known as "brute-force", uses empirical trial-and-error approaches and applied statistical methods targeting immediately working systems. The other, known as "perfectionist", uses theoretical approaches involving fundamental linguistic research to aim for long term solutions.

Optimism for MT lulled for a decade after the famous Automatic Language Processing Advisory Committee (ALPAC) report was published in 1966. The report pointed out that "there is no immediate or predictable prospect of useful machine translation." Instead of further investment in MT research, it recommended the development of machine aids for human translators and continued support of basic research in computational linguistics.

A decade later, however, MT revived with operational and commercial systems such as Systran. Rule-based models dominated the field until the end of the 1980s. These models essentially relied on linguistic rules such as rules for syntactic analysis, lexical transfer, syntactic generation, morphology, lexical rules, etc. During this period, researches attempted to develop advanced transfer systems building upon experience with earlier interlingua systems as well as to develop new kinds of interlingua. They investigated techniques and approaches from Artificial Intelligence.

The dominance of the rule-based approach waned in the late 1980s with the emergence of corpus-based approaches, which did not require any syntactic or semantic rules in text analysis or selection of lexical equivalents.

1.3 Corpus-Based Machine Translation

In 1988, a group of researchers at IBM developed the Candide system that used statistical methods as means of analysis and generation (Brown et al., 1988). They successfully demonstrated statistical translation by showing acceptable results: almost half the phrases translated were acceptable. With their successful demonstration, Statistical Machine Translation (SMT) rose to dominance.

Other researchers extended the IBM SMT to Phrase-Based SMT. Marcu and Wong (2002) studied a phrase-based joint probability model in which they learn translation between source n-grams and target n-grams. Others applied heuristics on the IBM word alignment to extract phrase translation pairs (Och and Hey, 2004; Koehn, 2004a). Chiang (2005) extended the IBM SMT to hierarchical phrase translation pair extraction in the HIERO system.

Nagao (1984) introduced another major corpus-based approach called Example-Based Machine Translation (EBMT) in the early 1980s, although experimentation on the approach did not begin until the end of 1980s. The underlying hypothesis of EBMT is that translation can benefit from using previously translated analogous examples. When EBMT is given an input sentence, it finds similar source sentences and their translations in an example database. After dealing with the differences in the similar examples, it comes up with hypothesis translations. EBMT systems are categorized by the forms of meta data with which they calculate similarity.

Lexical EBMT systems use the surface form of texts directly. Because finding very similar sentences in the surface form is rare, lexical EBMT systems typically use partial matches (Brown, 2000a,b; Phillips and Brown, 2009) or phrase unit matches (Veale and Way, 1997) [1]. To find hypothesis translations, they collect the translations of the matches for use in decoding. To increase coverage, lexical EBMT systems optionally perform generalization on the surface form to find translation templates.

Other EBMT systems use linguistic structures to calculate similarity. Some convert both source and target sentences in the example database [2] into parse trees, and when they are given an input sentence, they parse it and calculate similarity to the stored example parse trees. They then select the most similar source parse trees with their corresponding target trees to generate target sentences after properly modifying them by the differ-

[1] Sato (1992)'s system also uses surface form, but it uses a character-based similarity calculation.

[2] In this thesis, we use an example database and training set interchangeably.

ence (Sato and Nagao, 1990; Maruyama and Watanabe, 1992; Sumita and Iida, 1991; Al-Adhaileh and Tang, 1999; Aramaki and Kurohashi, 2004). Or they find source sub tree matches with their aligned target sub trees and combine the the target parts to generate target sentences (Quirk and Menezes, 2006). Others covert only the source side to make use of parse trees for similarity calculation (Langlais and Gotti, 2006; Liu et al., 2006). Andriamanankasina et al. (1999) converted sentences into Part-Of-Speech tags to measure similarity between input sentences and examples.

1.4 The CMU Example-Based Machine Translation system

In this thesis work, we used the CMU EBMT system which is a lexical EBMT system. The system is described in detail in Chapter 2.

1.5 Motivation

When we started this thesis work, we were looking for a new phrasal alignment algorithm, possibly a new translation unit and a way to integrate SMT alignment techniques into the EBMT system because those aspects had been less studied in the CMU EBMT system. The goal was to achieve a substantial improvement in the EBMT system by finding problems in the related components and developing reasonable solutions.

The CMU EBMT has been focusing on increasing the training corpus coverage over input sentences to be translated by using techniques such as word generalization (Brown, 2000a,b) rather than further developing accurate alignment. At the time, its approach to alignment was using a correspondence table for a training sentence pair which has a binary value for a source and target word pair representing alignment. The binary relationship was obtained from an automatically trained dictionary on the training set. At translation time, a heuristics-based aligner finds translations of partial source matches using the alignment information in the correspondence table. For this reason, the CMU EBMT system's alignment related components showed potential for improvement.

Additionally, Statistical Machine Translation (SMT) researchers have focused on finding more correct translations by finding more accurate word alignments (Brown et al., 1993; Och and Ney, 2000) and extracting a high quality phrase table from a training corpus.

Therefore, when we decided to improve alignment in the CMU EBMT system, SMT's word-to-word translation probability in the correspondence table in EBMT was essential so that EBMT assigns a more accurate probability as a weight to each corresponding word pair, leading to better translations.

Although one may suppose that the CMU EBMT system could use the translation table built by a Phrase-Based SMT system, the approach is not feasible because the CMU EBMT system needs to find phrasal alignments at translation time because it needs to find target phrases corresponding to arbitrary source matches. It is not realistic to build an SMT phrase table for EBMT phrasal alignment because the SMT phrase table that covers arbitrary source matches would be enormous when the size of training data is very large. This requirement led us to develop an algorithm that finds the most probable target phrase for an arbitrarily long input match. This algorithm, called Symmetric Probabilistic Alignment (SPA), finds the translation with a maximum symmetrized score based on a mathematical model rather than heuristics.

The initial SPA worked on a correspondence table of word-to-word translation probabilities rather than binary values. This assumed the availability of a probabilistic dictionary but not a reasonably large parallel corpus. For example, Mapadungun which is one of the indigenous languages in South America, has little parallel data with English. However, there exists a dictionary between those two languages [3]. Similarly, there may be languages between which a comparable corpus exists but not a parallel corpus. In this case, we can train a probabilistic dictionary but do not have a parallel corpus. However, where there are widely used language pairs for which a large parallel corpus is available, a probabilistic

[3] In this case, we need to assign a pseudo probability value to each translation pair. In our experiments, we simply use a word probability dictionary obtained from an SMT word alignment algorithm.

word translation dictionary and word alignment information drawn from it is also available. We extended SPA to use the word alignment information, which was then used in finding a possible target phrase range or in finding non-contiguous alignment.

Often, we observe that tokens which do not have translational equivalents cause a problem in translation. When they are in the source side, they insert irrelevant target words in the translation that were automatically found in a training phase. When they are in the target side, they are typically missing or inserted inconsistently. A very simple example may be a Korean phrase ‘na neun’ literally meaning ‘I NOMINATIVE’. When it is translated into English, ‘na’ is translated to ‘I’ and ‘NOMINATIVE’ is translated to an irrelevant token [4]. One way to overcome this problem is to consider ‘na neun’ a single translation unit. By having ‘I’ as the translation of ‘na neun’, we can translate it correctly. We investigate this problem with linguistically motivated phrases, chunks in our EBMT system.

Analyzing sentences into their chunks instead of SMT style phrases potentially aids a translation system in a few ways. With fewer translation units per sentence, overall distortion decreases (or rather, the distortion has been reduced to a local and global component, and the local reordering is accessed by rote). Hence, less noise is to be expected from the mathematical modeling techniques. For example, when we perform alignment on an English sentence “I go to the park with my dog .” and its Korean translation “na neun na eui gae reul derigo gongwon e ganda [5].” , we have 9 English words and 11 Korean words to align and the second English word ‘go’ should be aligned to the 10th Korean word ’ganda’. But if we chunk them and perform alignment on the chunked sentences “[I] [go] [to the park] [with my dog] [.]” and “[na neun] [na eui gae reul derigo] [gongwon e] [ganda] [.]”, we have 5 English chunks and 5 Korean chunks and ‘go’ and ‘ganda’ are just 3 chunks away. Obviously the chunked sentences are easier for alignment because there is less distortion and higher correspondence. Another advantage is that we can to some degree systematically translate untranslatable tokens that exist only on one side. For example, when we translate an English sentence into Korean, a word-to-word translation systems cannot produce a nominative case marker in Korean unless rules are given by human experts or

[4] In our observation, ‘NOMINATIVE’ is often translated to ’the’.

[5] The Korean tokens corresponds to “I NOMINATIVE I of dog ACCUSATIVE with park to go .”

the systems “hallucinate” markers and use language modeling to guess whether or not the case marker should in fact be present. This ability to generate lexical tokens from their presence in chunks is particularly useful for function words. Otherwise potentially unrelated function words in two languages are very often aligned even if they are not translational equivalents (Fossum et al., 2008). If this kind of alignment is restricted by chunks, it helps not only the word alignment but also the phrase alignment derived from the word alignment.

A phrasal aligner such as SPA may also find the correct chunk translation. It could find the correct chunk translation answer as the best translation, have it in the top-N list, or prune it out. In this case, the translation system needs a good mechanism to make sure that SPA returns the correct target chunk and the decoder picks it correctly with the help of a language model. However, because chunk alignment finds a single target chunk given a source chunk, it can encourage the system to use the correct chunk translation.

For this reason, we investigated machine translation with chunks as basic units. We first developed a chunk alignment algorithm and evaluated it. Then we used the aligned chunk translations in the CMU EBMT system to improve system performance. Finally we investigated whether we could improve a Phrase-Based SMT system by adding chunk translation pairs to its phrase table.

1.6 Thesis hypotheses

Through this thesis work, we strive to validate the following hypotheses.

First, Symmetric Probabilistic Alignment (SPA) will improve the CMU EBMT system. With a more accurate phrase level alignment than the existing heuristic aligner, the EBMT system will perform better.

Second, using state-of-the-art word alignment information in SPA will help SPA output better target phrases. The external word alignment will be useful not only for determining a target range in which SPA finds translation candidates but also for providing its own phrasing as a good translation candidate.

Third, non-contiguous SPA will be better for translation than contiguous SPA. With accurate word alignment, non-contiguous SPA will have higher precision and it will lead to better translation.

Fourth, our chunk alignment will be better than its baselines: chunk alignment by state-of-the-art word alignment algorithm that regards chunks as basic units and chunk alignment in which a source chunk and a target chunk are aligned when there is any aligned word pair between them. Our aligner uses both word and chunk statistics for alignment, which will lead to higher chunk alignment accuracy.

Fifth, our chunk alignment method will help find high quality chunk pairs. Adding these pairs to a Phrase-Based Statistical Machine Translation (PBSMT) phrase table will improve a PBSMT system.

Sixth, iteratively performing word alignment and chunk alignment will improve both alignments. By using chunk boundary constraints in word alignment, word alignment quality will improve and by using improved word alignment, chunk alignment will improve.

Finally, by using chunks as basic translation units with the help of a lexical model and by giving more credit to high-accuracy chunk translations, we can surpass the lexical model in translation quality.

Chapter 2

The CMU EBMT System

2.1 The CMU EBMT system in a nutshell

Because our intent is improving the CMU EBMT system, this chapter describes the CMU EBMT system at the time we began our thesis work. Later chapters will describe changes to the system, as they were made within the experiment.

Figure 2.1 shows a diagram of the CMU EBMT system. The system is a lexical EBMT system, meaning that it calculates similarity on the surface form of texts (Brown, 1996, 2004). In other words, given an input sentence to be translated, the system finds similar sentences in the surface form. In the system, the similarity calculation was implemented by finding contiguous source word matches in a stored example database. For each match in a sentence pair, the system finds its translation phrase using a word-to-word correspondence table, in which all the word-to-word mappings have a binary correspondence value indicating whether they are translations or not. In the rest of this chapter, we describe the detailed role of each component in Figure 2.1 in training and run (translation) time.

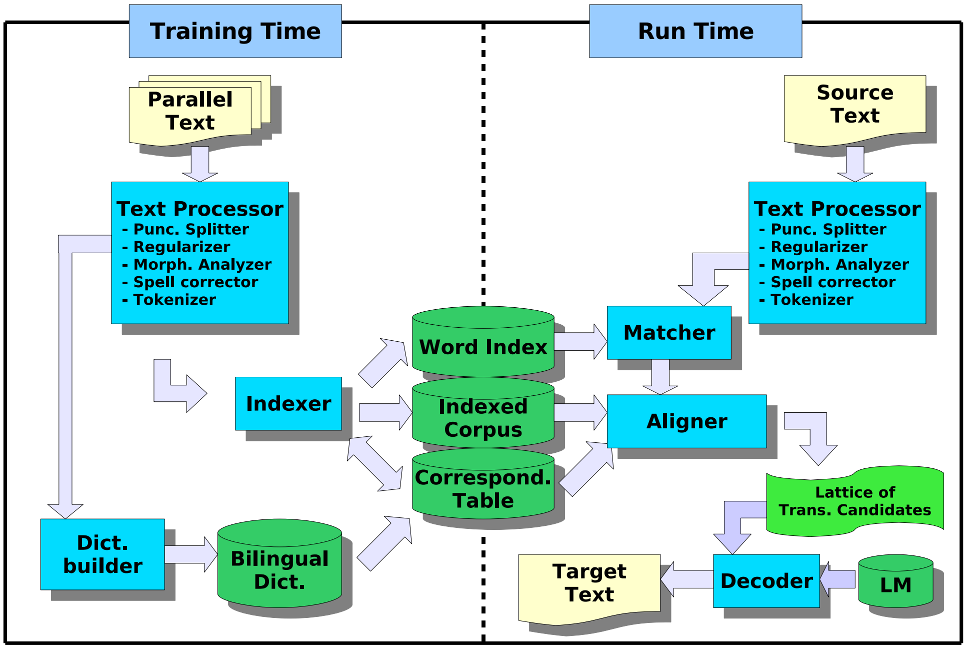


Figure 2.1: The CMU EBMT system

2.1.1 Training

During training time, the system transforms the data for efficient matches and builds a dictionary and correspondence tables to be used in translation.

Pre-processing: The input for the training stage is a parallel corpus which is a list of translation sentence pairs. Once the system is given a parallel corpus, it performs pre-processing on both language sides of the data.

- The *Punctuation Splitter* splits punctuation marks from words. It can take abbreviations as input for each language and leave them unchanged.
- The *Regularizer* transforms the form of words. For example, "'m" in "I'm" can be transformed into "am" after detaching it from "I," so that "I'm" can be matched for an input "I am".

- The *Morphological Analyzer* may be used for a morphologically rich language for better word match and higher word occurrences.

- The *Spell Corrector* can be used to correct misspelled words (if applicable) as described by Hogan (1998).

- The *Tokenizer* decides whether a series of tokens should be split. For example, it will attach "AT", "&" and "T" to have "AT&T" as a unit.

Dictionary Building: Next, a *dictionary builder* collects co-occurrence statistics for source and target word pairs. Using a pre-specified threshold for the co-occurrence statistics, it selects co-occurring word pairs and adds them into a dictionary.

Correspondence Table Building: The system then builds a *correspondence table* for each sentence pair in which every source and target word pair has a binary relationship. If a pair is found in the dictionary built in the previous step, it assigns a binary value "1" to the pair to indicate that they correspond to each other in the sentence pair as translations. Otherwise, the pair is given "0". Depending on the similarity of the language pair, the system may apply pruning to remove out lier correspondents. For example, in a Spanish-English translation sentence pair, an acceptable target word range of a source word is determined by finding the earliest and latest word positions of the first best and the third best target words and expanding them by N (normally, 2) words on both the left and right to allow for word-order variations (Brown, 1997) [1].

Corpus Indexing: The system assigns each sentence pair a unique ID (sequential integers were chosen for efficient retrieval), which is then stored.

Word Indexing: As mentioned earlier, the system can find contiguous source matches of previously unspecified lengths. To support this function, it builds an index database on the training set so that given an input sentence to be translated, it finds training sentence pairs whose source side includes a fragment of the input. The Burrows-Wheeler Transform (BWT) is used to support efficient lookups in a scalable system (Brown, 2004).

[1] Note that our dictionary was automatically built based on co-occurrence statistics and may have noisy translations, which consequently lead to noisy correspondence.

2.1.2 Translation

During translation time, the system uses the data prepared during the training time.

Pre-processing: When an input sentence for translation arrives, the system performs the same pre-processing as it did for the source side of the training set.

Matcher: After the input is pre-processed, the *Matcher* finds the longest match from each source word position and its sub strings starting with the same position. For example if the *Matcher* found “word1 word2 word3”, it also finds “word1 word2” and “word1”

Because some n-grams (including unigrams) appear very often, the system can set a limit on the number sentences to include matches. For example, the English word “I” appears so frequently that it does not make good sense to retrieve all the “I”s throughout the entire corpus. Instead, the system will use a subset of the entire matched sentences using a specified limit on the number of matched sentences [2]. If this limit is too large, the speed of the system will decrease. However if the limit is too small, the system will only find a small number of translation candidates from the retrieved sentence pairs.

Aligner: For a source match, the system asks the *Aligner* to find its translation. The input to the *Aligner* consists of the matched source phrase, a sentence pair that includes the matched source phrase on the source side, and the correspondence table of the sentence pair. First, the *Aligner* finds the shortest and longest contiguous target phrases that include the correspondent target words from all the matched source phrase words. Next, for each substring of the longest contiguous target phrase that also includes the shortest one, it calculates an alignment score based on heuristic functions. Finally, it returns the single target substring that has the highest alignment score as the best translation of the source match.

The system puts best N target translation of each source match in a lattice with the alignment score, where N is a configurable parameter for the maximum number of translations for each source match.

Decoder: Finally, the system invokes the *Decoder* to find the best possible translation

[2] In the experiments performed in this thesis, we set the limit to 2,000.

hypothesis. The *Decoder* uses a beam and can control the size of the beam with a specified beam size and hypothesis score ratio from the best hypothesis score.

The hypothesis score is calculated from the alignment score and other EBMT feature scores. The EBMT feature scores combined during decoding include:

- **Language Model Score** is the probability of the hypothesis sentence calculated using a language model
- **Arc Weight** combines engine-specific weights for each engine that contributes an identical source/target pair to the lattice plus a bonus for multiple engines contributing the same pair.
- **Score** is the engine's score for the quality of the translation pair, or, if multiple identical arcs were merged, the average of the scores.
- **Verbosity Penalty** sets the strength of the penalty for having output that varies from the expected length.
- **Reorder Penalty** is the amount used to scale the total number of re-orderings performed on a path through the lattice.

These are combined using the linear interpolation method. The experiments encompassed by this thesis were conducted using only the features above. Although other features exist, they were disabled for the experiments in this thesis.

2.1.3 Difference from Phrase-Based Statistical Machine Translation systems

Like our lexical EBMT system, a typical Phrase-Based SMT (PBSMT) system such as Moses (Koehn et al., 2007), also finds contiguous partial source matches and their translations in the pre-built phrase translation table during training time. The difference is that PBSMTs build a phrase translation table during the training time and use that to find source phrase matches and their translations. Thus given an input sentence, they cannot find translations for an arbitrary source match. Their source matches are restricted to the source phrases in the pre-built table.

However the CMU EBMT builds a dynamic phrase table per sentence during translation time. This means that it asks the phrasal aligner to find translation candidates of an arbitrary source match during the translation time. In this aspect it is very similar to the CMU SMT system that uses PESA (Vogel, 2005) phrasal aligner during translation time. PESA was developed concurrently with SPA.

2.1.4 The problems with the current system

In Figure 2.1, we see room for improvement in the correspondence table and the aligner (highlighted in Figure 2.2).

In considering improvements to the correspondence table of word alignments for a sentence pair, we recognized that the current correspondence table is limited by aligning words only using only a dictionary and heuristic-based pruning. It also uses binary values even though word translation probabilities can better represent the strength of relationship between a source word and a target word.

The aligner uses only heuristic-based functions to calculate the alignment score. Examples of the heuristic-based functions include the number of the relationship “1” in the correspondence table, the target phrase length discrepancy from the expected target length calculated using the source phrase length and the source and target sentence length ratio, the ratio of the source phrase length in the source sentence, etc.

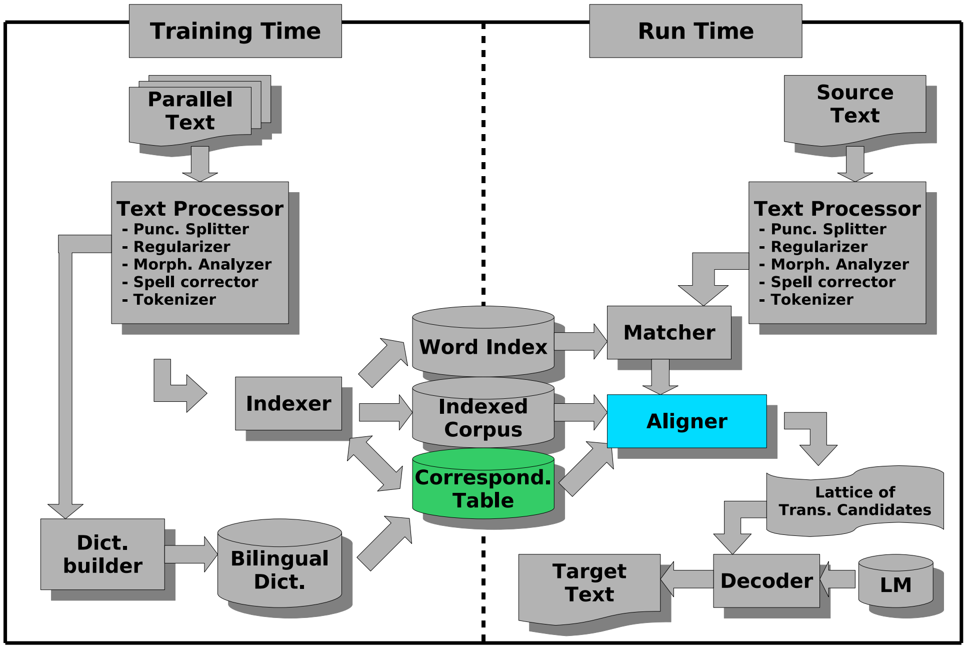


Figure 2.2: Components of the CMU EBMT system to be improved.

Chapter 3

Symmetric Probabilistic Alignment

In this chapter, we describe our basic Symmetric Probabilistic Alignment (SPA) algorithm and the restrictions we applied to it for improvement.

We performed evaluations to measure phrasal alignment accuracy and translation quality. For alignment accuracy evaluation experiments, we obtained a small hand-aligned corpus for English-Chinese and French-English pairs. For translation, we drew a small amount of data from French-English Canadian Hansards corpus and annotated it with phrasal alignments using SPA. The annotated corpus was used as a training set from which the EBMT system found partial matches and their alignments for input sentences.

3.1 Related work

There has been much work in the field of word alignment because it is such an important task in corpus-based machine translation. Many methods and algorithms have been developed by various machine translation groups. Some used heuristic-based methods, others, pure statistical approaches, and still others, linguistic knowledge in alignment.

Smadja et al. (1996) and Melamed (2000) have used similarity functions between two languages. Variants of the Dice coefficient, Dice (1945), have frequently been used to calculate similarity by obtaining a matrix that includes association scores between each pair of a source word and a target word at different positions for each sentence pair. Melamed (1997) applied a constraint on this score to overcome indirect associations so as to avoid the association between two words that appear together very frequently but do not have a translation relationship.

At the IBM T.J.Watson Research Center in the early 1990s, Brown et al. (1993) developed several alignment models for use with the EM algorithm, which are now commonly called IBM model 1, 2, 3, 4, and 5 and intended to provide increasingly more accurate models of the translation process. In their noisy channel model, the translation model can be written as a combination of alignment probability and translation probability:

$$\begin{aligned} Pr(\mathbf{f}|\mathbf{e}) &= \sum_{\mathbf{a}} Pr(\mathbf{f},\mathbf{a}|\mathbf{e}) & (3.1)\\ &= \sum_{\mathbf{a}} Pr(m|\mathbf{e}) \prod_{j=1}^{m} Pr(f_j, a_j|a_1^{j-1}, f_1^{j-1}, m, \mathbf{e}) & (3.2)\\ &= \sum_{\mathbf{a}} Pr(m|\mathbf{e}) \prod_{j=1}^{m} Pr(a_j|a_1^{j-1}, f_1^{j-1}, m, \mathbf{e}) Pr(f_j|a_1^{j}, f_1^{j-1}, m, \mathbf{e}) & (3.3)\end{aligned}$$

for an English string $\mathbf{e} = e_1^l \equiv e_1 e_2 ... e_l$, a French string $\mathbf{f} = f_1^m \equiv f_1 f_2 ... f_m$ and their alignment $\mathbf{a} = a_1^m \equiv a_1 a_2 ... a_m$ [1].

Model 1 assumes that for a source word position, all connections to target word positions are equally likely (i.e., all the possible alignments are equally likely). The alignment probability is

$$\prod_{j=1}^{m} Pr(a_j|a_1^{j-1}, f_1^{j-1}, m, \mathbf{e}) = (l+1)^{-1} \qquad (3.4)$$

In Model 2, they have a more realistic assumption that the probability of a connection between a source position and a target position depends on the positions it connects and on the lengths of the two strings(i.e., a source string and the corresponding target string in

[1] English and French were the original target and source languages in IBM's Candide project, but ”e” and ”f” are now commonly used in SMT regardless of the actual languages. In this thesis, we use the same notations for source and target languages.

a parallel corpus) The connection between positions is known as distortion. The alignment probability in Model 2 is

$$d(a_j|j, m, l) \tag{3.5}$$

In Model 3, Brown and his colleagues introduce the concept of word fertility. First they choose the number of French words that are connected to an English word and then follow the procedure for Model 2. Because they choose the number of French words associated with a given English word, the direction of the distortion model is reversed this time:

$$d(j|i, m, l) \tag{3.6}$$

Model 4 is designed to model the fact that an English string is often translated into French as a unit. They define the *center* of an English cept to be the ceiling of the average value of the positions in the French string of the words. Model 5 is very much like Model 4 except that it is not deficient. A deficient model can choose the same target position repeatedly for the target words given different source words and could result in too many empty target positions. In these models, accurate parameter estimation is the key point for improving the performance of the models and they used EM algorithms to estimate parameters. Because EM algorithms converge to local maxima, they use the previous model's parameters as the initial parameter values to achieve better performance.

Vogel et al. (1996) have used Hidden Markov Model (HMM) in alignment to take into account that the previous French word's alignment restricts the next French word's alignment position. They assume that there is a first-order dependence for the alignments and that the lexicon probability depends only on the word at a given position.

$$\begin{aligned} Pr(f_j, a_j|a_1^{j-1}, f_1^{j-1}, m, \mathbf{e}) &= p(f_j, a_j|a_{j-1}, \mathbf{e}) && (3.7) \\ &= p(a_j|a_{j-1})p(f_j|e_{a_j}) && (3.8) \end{aligned}$$

Yamada and Knight (2002) use syntactic parse trees on the English side and plain text on the French side. To model this pair, they use three operations: *reordering*, *insertion* of French words, and *translation* of English words. They raise the prospect of training an SMT system using syntactic information for both languages.

Many algorithms have been designed to go beyond word-to-word models as well.

Wu (1997) studied inversion transduction grammar formalism for a parallel corpus. The goal is to generate a pair of strings in two languages simultaneously using a bilingual probabilistic context-free grammar. This naturally aligns not only words but also phrases. Sub trees in a parse tree are word/phrase alignments for the sentence pair.

Marcu and Wong (2002) studied a phrase-based joint probability model that generates and orders phrases in both languages employing a number of concepts. Their model can be described by formula

$$p(\mathbf{e},\mathbf{f}) = \sum_{C \in \mathcal{C} | L(\mathbf{e},\mathbf{f},C)} \prod_{c_i \in C} [t(\overrightarrow{e}_i, \overrightarrow{f}_i) \times \prod_{k=1}^{|\overrightarrow{f}_i|} d(pos(\overrightarrow{f}_i^k), pos_{cm}(\overrightarrow{e}_i))] \qquad (3.9)$$

where $L(\mathbf{e},\mathbf{f},C)$ means that a sentence pair **e** and **f** are obtained by permuting the phrases $\overrightarrow{e}_i$ and $\overrightarrow{f}_i$ that characterize all concepts $c_i \in C$, $pos(\overrightarrow{f}_i^k)$ denotes the position of kth word in the phrase $\overrightarrow{f}_i$ and $pos_{cm}(\overrightarrow{e}_i)$ denotes the position of the center of the phrase $\overrightarrow{e}_i$. They show a significantly better score than IBM Model 4, which is word-based, but their training for n-gram phrases translation table is computationally intractable for even moderate values of n due to its size.

To overcome the problems of word-to-word alignment, an alignment template by Och and Hey (2004) and phrase extraction by Koehn (2004a) were studied. They note that word-to-word alignment is limited by each French word being aligned to only one English word in the IBM models. Therefore they train IBM Model 4 for $P(\mathbf{e}|\mathbf{f})$ and $P(\mathbf{f}|\mathbf{e})$ and take the intersection of the two alignments to get a high-precision alignment as a starting point. They then explore the union of the alignments and expand the intersection by adding an alignment point that aligns a word which currently has no alignment. After building a matrix of alignments, they extract consistent phrase pairs.

Vogel (2005) and Zhao and Vogel (2005) treat phrasal alignment as a sentence splitting problem. Using a lexicon, they locate a target phrase where the lexicon probability is optimal. Then they extend it to use a fertility model to better estimate the target phrase length.

A hierarchical phrasal alignment was studied by Chiang (2005) and Chiang et al. (2005). He and his colleagues train a synchronous context-free grammar from a word-aligned bilingual corpus to learn global reordering.

Simard et al. (2005) built a phrase-based statistical translation model based on non-contiguous phrases to better take into account additional linguistic phenomena that contiguous phrase-based model cannot capture. To produce a non-contiguous phrase pair library, they tested two strategies: combining contiguous phrase pairs occurring in the same sentence, which were found by the *Refined Method* described in Och and Ney (2003) and combining cepts found by a matrix factorization in Goutte et al. (2004).

A very different tack was taken by Veale and Way (1997) in the *Gaijin* Example-Based Machine Translation system and its successors. They first find constituent-based chunks mono-lingually and then attempt to match corresponding chunks between the two languages. Chunk boundaries are found by applying Green's Marker Hypothesis (Green, 1979) using hand-written sets of marker words such as determiners and prepositions.

Our own previous work on alignment, Symmetric Probabilistic Alignment (SPA) Kim et al. (2005), found phrase-to-phrase mappings by bootstrapping word-to-word translation probabilities to determine the target-language phrase with the best bidirectional alignment score for an arbitrary source-language phrase. It can find a target phrase for an arbitrary source phrase. The algorithm is described in Section 3.2 in detail. As previously mentioned, alignment is fundamental to data-driven machine translation approaches. In this chapter, we describe our sub-sentential alignment method that finds target fragments for an arbitrary source match in our EBMT system.

3.2 Algorithm

3.2.1 Basic algorithm

In sub-sentential alignment, mappings are produced between words or phrases in the source language sentence and those words or phrases in the target language sentence that best express their meaning.

An alignment algorithm takes as input a bilingual corpus consisting of corresponding sentence pairs and strives to find the best possible alignment in the second for selected n-grams (sequences of n words) in the first language. The alignments are determined based on a number of factors, including a bilingual dictionary (preferably a probabilistic one), the position of the words, punctuation, invariants (such as numbers), and so forth.

For our baseline algorithm, we make the following simplifying assumptions, each of which we then relax:

1. A fixed bilingual probabilistic dictionary is available.

2. Contiguous fragments (word sequences) of source language text are translated into contiguous fragments in the target language text.

3. Fragments are translated independently of surrounding context.

Our baseline algorithm is based on maximizing the probability of bi-directional translations of individual words between a selected n-gram in the source language and every possible n-gram in the corresponding paired target language sentence. The reason why we use the probability of bi-directional translations is that we are more convinced when both side's fragments agree that the other sides' fragments are their translations. For example, given a source fragment f_i^j, assume that the two target fragments e_k^l and e_n^o are equally probable 'best' translations of f_i^j. If we consider opposite directional translations and find that e_k^l's the most probable translation is f_i^j and e_n^o's the most probable translation is f_p^q ($i \neq p$ or $j \neq q$), we will choose e_k^l as the translation of f_i^j.

No positional preference nor length preservation assumptions are made. That is, an n-gram may translate to an m-gram, for any values of n or m bounded by the source and target sentence lengths, respectively. Finally, we introduce a small positive ”smoothing value” ϵ to avoid singularities (i.e. avoiding zero-probabilities for unknown words or words never before translated in a way consistent with the dictionary).

Suppose that we are given a pair of aligned sentences **F** of length K and **E** of length L where a source sentence **F** is

$$\mathbf{F} = f_1, ..., f_{i+1}, ..., f_{i+k}, ..., f_K \tag{3.10}$$

and the corresponding target language sentence $\mathbf{E}$ is

$$\mathbf{E} = e_1, ..., e_{j+1}, ..., e_{j+l}, ..., e_L \tag{3.11}$$

and calculating the translation probabilities between a source fragment f_{i+1}^{i+k} and target fragments in $\{e_{j+1}^{j+l}\}$.

Then the fragment we try to obtain is the target fragment $\overline{\mathbf{e}} = e_k^l$ with the highest probability of all possible fragments of $\mathbf{E}$ to be a mutual translation with the given source fragment, or

$$\begin{aligned}
\overline{e} &= \text{argmax}_{\mathbf{e}} Score_{\mathbf{e}} & (3.12)\\
&= \text{argmax}_{\mathbf{e}}(p(f_{i+1}^{i+k} \leftrightarrow e_{j+1}^{j+l})) & (3.13)\\
&= \text{argmax}_{\mathbf{e}}(p(f_{i+1}, ..., f_{i+k} \leftrightarrow e_{j+1}, ..., e_{j+l})) & (3.14)\\
&= \text{argmax}_{\mathbf{e}}((\prod_{p=1}^{k} \max(\max_{q=1}^{l} p(e_{j+q}|f_{i+p}), \epsilon))^{\frac{1}{k}} & (3.15)\\
&\quad \times (\prod_{q=1}^{l} \max(\max_{p=1}^{k} p(f_{i+p}|e_{j+q}), \epsilon))^{\frac{1}{l}}) & (3.16)
\end{aligned}$$

Here and in the following sections for algorithm description, we use $\mathbf{e} = e_{j+1}^{j+l}$ for the target candidate fragment $\mathbf{e}$.

In the above equation, (3.15) shows the unidirectional score calculation from source to target, and (3.16) shows the unidirectional score calculation from target to source. So, (3.15) and (3.16) together calculate the symmetric probabilistic alignment score.

In this algorithm, given a source phrase, we check $\frac{L\times(L+1)}{2}$ fragments where L is the target language length because we will check L 1-word-long fragments, $L-1$ 2-word-long fragments, and so on.

3.2.2 Untranslated word penalty

In our basic algorithm, we calculated a symmetric probabilistic alignment score but did not count how many words in the counterpart fragment are actual translations for the given fragment words. Instead we prefer an alignment that has more actual translations in the counterpart fragment. For example, for a given source fragment $\mathbf{f} = f_{i+1}^{i+k} = f_{i+1}, ..., f_{i+k}$ and a given candidate target fragment $\mathbf{e} = e_{j+1}^{j+l} = e_{j+1}, ..., e_{j+l}$, if all source words in $\mathbf{f}$ are translated into a single target word in $\mathbf{e}$, and if all target words in $\mathbf{e}$ are translated into a single source word in $\mathbf{f}$, this alignment is not desirable and should be penalized.

So we will penalize the alignment score according to the ratio of $\frac{\#(translations)}{|fragment|}$. A modified formula would be

$$\begin{aligned} Score_{\mathbf{e}} &= P(f_{i+1}^{i+k} \leftrightarrow e_{j+1}^{j+l}) \\ &= P(f_{i+1}, ..., f_{i+k} \leftrightarrow e_{j+1}, ..., e_{j+l}) \\ &= (\prod_{p=1}^{k} \max(\max_{q=1}^{l} p(e_{j+q}|f_{i+p}), \epsilon))^{\frac{1}{k}} \times (R_{\mathbf{e}})^{\alpha} \\ &\quad \times (\prod_{q=1}^{l} \max(\max_{p=1}^{k} p(f_{i+p}|e_{j+q}), \epsilon))^{\frac{1}{l}} \times (R_{\mathbf{f}})^{\alpha} \end{aligned} \tag{3.17}$$

where $R_{\mathbf{p}} = \frac{\#\ of\ actual\ translation\ words\ in\ the\ fragment\ \mathbf{p}}{\#\ of\ potential\ translation\ words\ in\ the\ fragment\ \mathbf{p}}$, and $\alpha \geq 1$. In this formula, when $R_{\mathbf{p}}$ is less than 1, it reduces $Score_{\mathbf{e}}$ and, as a result, penalizes the score. In the previous example, $R_{\mathbf{p}} = \frac{1}{l}$ and it obviously reduces $Score_{\mathbf{e}}$ when $l > 1$.

3.2.3 Length penalty

The ratio of target fragment (*n*-gram) lengths and source fragment (*m*-gram) lengths should be comparable to the length ratio of the target sentence and source sentence lengths, though certainly variation is possible. Therefore, we generate a penalty function to the alignment probability that increases with the discrepancy between the ratios as n/m is compared to the target/source sentence length ratio $\frac{L}{K}$.

If the length of the source language fragment is k, the length of a target language fragment under consideration is l, the dynamic sentence length ratio is $\frac{L}{K}$ given the source language sentence $\mathbf{F}$ and its corresponding target language sentence $\mathbf{E}$ in Section 3.2.1, the expected target fragment length is then given by $\hat{l} = k \times \frac{L}{K}$. Further defining an allowable length difference $LD_{allowed}$, our implementation calculates the length penalty LP as follows:

$$LD_{allowed} = LD_{constant} \times \frac{L}{|\mathbf{E}|_{average}} \tag{3.18}$$

$$LP = \min((\frac{|l - \hat{l}|}{LD_{allowed}})^4, 1) \tag{3.19}$$

where $|\mathbf{E}|_{average}$ means *the average target sentence length in the training corpus*.

We wanted to ignore target candidate fragments that have larger differences than $LD_{allowed}$ and to give an increasingly larger penalty to the $LD_{allowed}$-satisfying target candidate fragments as they have larger differences. For equation (3.19), the 4th power was the one that gave us the best experimental results among the powers from 2 through 6.

The score for a fragment including the penalty function is then:

$$Score_{\mathbf{e}} \leftarrow Score_{\mathbf{e}} \times (1 - LP) \tag{3.20}$$

Note that, as intended, the score is forced to 0 when the length difference $|l - \hat{l}| > LD_{allowed}$.

3.2.4 Distance penalty

Closely related languages, such as French and English, tend to have more similar word order than more distantly-related languages such as Korean and English. In the former case, this results in greater phrase order similarity and, consequently, similar phrase positions.

In such a close language pair, we introduce a distance penalty [2] to increasingly penalize the alignment score of any candidate target fragment as it moves away from the expected position range. Our distance penalty follows the same calculation method as in section 3.2.3. First, we calculate the expected center $\hat{C}$ of the candidate target fragment using the

[2] Our distance penalty is conceptually different from the distortion penalty in SMT systems because it assumes that a target fragment in a target sentence should be in a position proportional to the source fragment position in the source sentence. The distortion penalty in SMT systems is defined by a probability that a source position and a target position are connected.

center of the source fragment $C_{\mathbf{f}}$ and the dynamic sentence length ratio $\frac{L}{K}$

$$\hat{C} = C_{\mathbf{f}} \times \frac{L}{K} \tag{3.21}$$

Then we calculate $DD_{allowed}$, *the dynamic allowed distance difference of the center*, using a constant limit value $DD_{constant}$ and the dynamic sentence length ratio $\frac{L}{|\mathbf{E}|_{average}}$ where $|\mathbf{E}|_{average}$ is the average target sentence length in the training corpus.

$$DD_{allowed} = DD_{constant} \times \frac{L}{|\mathbf{E}|_{average}} \tag{3.22}$$

Given $DD_{allowed}$, we calculate the distance penalty DP as follows:

$$DP = \min((\frac{|C_{\mathbf{e}} - \hat{C}|}{DD_{allowed}})^4, 1) \tag{3.23}$$

where $C_{\mathbf{e}}$ is the actual center of the target fragment $\mathbf{e}$ being processed.

As we did in Section 3.2.3, we want to ignore target candidate fragments which have larger differences than $DD_{allowed}$ and to give larger penalties to the $DD_{allowed}$-satisfying target candidate fragments as their differences increase. For equation (3.23), as in the length penalty calculation, the 4th power was the one that gave us the best experimental results among the powers from 2 through 6.

The score for a fragment including the penalty function is then:

$$Score_{\mathbf{e}} \leftarrow Score_{\mathbf{e}} \times (1 - DP) \tag{3.24}$$

Note that, as intended, the score is forced to 0 when the length difference $|C_{\mathbf{e}} - \hat{C}| > DD_{allowed}$.

It may in fact be possible to usefully apply the distance penalty to language pairs in which the language pairs have a very dissimilar word order, provided we can determine or estimate a positional mapping between the sentences in a pair, and then use the distance with respect to this mapping.

3.2.5 Anchor context

If the words adjacent to the source fragment and the candidate target fragment are translations of each other, we expect that this alignment is more likely to be correct because adjacent source words are usually aligned to adjacent target words and, in this case, an alignment of adjacent words adds supporting evidence to the alignment we are considering. We combine $Score_{\mathbf{e}}$ with the anchor context alignment score $AnchorScore_{\mathbf{e}}$ by a linear weighted combination in log space,

$$\begin{aligned} AnchorScore_{\mathbf{e}} &= (P(f_i \leftrightarrow e_j) \\ &\quad \times P(f_{i+k+1} \leftrightarrow e_{j+l+1}) \\ &\quad \times P(f_i \leftrightarrow e_{j+l+1})) \\ &\quad \times P(f_{i+k+1} \leftrightarrow e_j))^{\frac{1}{4}} \end{aligned} \tag{3.25}$$

$$Score_{\mathbf{e}} \leftarrow (Score_{\mathbf{e}})^{\lambda} \times (AnchorScore_{\mathbf{e}})^{1-\lambda} \tag{3.26}$$

Empirically, we found this combination gives the best score when $\lambda = 0.8$ for both French-English and English-Chinese and it gives a better result than

$$Score_{\mathbf{e}} \leftarrow \lambda \times Score_{\mathbf{e}} + (1-\lambda) \times AnchorScore_{\mathbf{e}} \tag{3.27}$$

3.3 Evaluation

3.3.1 Alignment evaluation

Data We tested our alignment method on a set of French-English sentences taken from the Canadian Hansard corpus and on a set of English-Chinese sentences taken from Xinhua news agency. French and English are chosen as an easy pair because they have very similar word order while English and Chinese are chosen as a difficult pair because the word order difference and the sentence length difference are the most evident.

For French-English, we had 91 human word-aligned sentence pairs, and from that, we generated 12466 3-8 words long contiguous source fragments.

For English-Chinese, we had 3 sets of 366 human aligned sentence pairs with the same data but are aligned by different people (The sets are named A, B and C). In addition to the three sets, we had 20 more human aligned sentence pairs aligned by another person. So, for the alignment evaluation, we picked one of the three sets - A was picked in this experiment - and added it to the other 20 sentences to make a 386 human aligned sentence pair set and 27,286 3-8 words long source fragments. And later we used the 3 sets to see how reliable human alignments are by evaluating each set against the other two.

For these experiments, we pre-processed the data. We segmented the Chinese data into words, and expanded the contractions in the French and English data. We separated the punctuation in the data in all three languages. For Chinese segmentation, we used lrSegmentor by Zhang.

Evaluation metric For the human-aligned data, we compared the results of our algorithm to the human alignments. Although the latter may not be perfect and are sometimes non-unique, they provide the only answer key available for repeatable tests. As metrics, we use *precision*, *recall* and F_1 (the harmonic mean of precision and recall). Since *precision* and *recall* cannot be used alone to measure the performance of the alignment methods, we use F_1 values to measure the performance and to compare the alignment methods. In other words, we use F_1 to measure the performance in both terms of both *precision* and *recall*.

We calculate precision, recall and F_1 based on answer position overlaps. Let us suppose that the position sequence of our (machine) answer fragment is $p_1, p_2, ..., p_k$ and the position sequence of the correct answer (human) fragment is $hp_1, hp_2, ..., hp_l$. Note that the correct (human) answer fragment may be non-contiguous, but the combination of SPA and EBMT to date is only capable of using the best *contiguous* target m-gram alignment it can find. Given that $o = count(p0_i)$ and $p0_i$ is p_i, which is not aligned in the human answer, we compute the recall R and precision P as follows:

$$R = \frac{|\{hp_i\} \cap \{p_j\}|}{l} \tag{3.28}$$

$$P = \frac{|\{hp_i\} \cap \{p_j\}|}{k - o} \tag{3.29}$$

To obtain an average alignment score for evaluation, we

- generated all the possible source language sentence fragments lengths 3 through 8 from the human-aligned data [3];

- aligned those fragments by means of our algorithm; and

- calculated the metrics given above by comparison with the human-aligned answers.

Baselines To better understand the alignment results we obtain for a given language pair (and corpus), we introduce the following as baselines: “random result,” “positional result,” and “oracle result.”

The “random result” is a randomly chosen target fragment, regardless of the source fragment, constrained to be of a length corresponding to the source fragment normalized by the length ratio of the source and target sentences.

The “positional result” is a target fragment whose position in the target language most closely matches the position of the source fragment. We calculate the target fragment’s start and end positions using the source fragment’s start and end positions as well as the length ratio of the source sentence and target sentences. In particular, if the source sentence is of length n and the target sentence of length m, we expect source position i to correspond to target position j where $j \simeq i \times \frac{m}{n}$.

The “oracle result” is the best contiguous target fragment extracted from the human alignments. To get the oracle result, we first get human alignments for the sentence pairs that will be used to evaluate our algorithm. Then we choose the fragment that has the largest harmonic mean value among human alignment fragments and whole fragment. Notice that the human alignment may not be contiguous, therefore “oracle alignment” represents the best that our algorithm could possibly perform.

Comparison with the state-of-the-art alignment We also included the IBM Model 4 (“IBM4”) alignment accuracy to evaluate the status of SPA compared to the state-of-the-

[3] In this experiment, we focused on the performance of systems for matches longer than 2 because shorter ones can be covered by the EBMT dictionary.

art model [4].

Finally we combine the results of SPA and IBM4. We set a threshold score for SPA and combined SPA and IBM4 results by substituting IBM4 results with SPA results that have higher alignment score than the threshold("COMB") [5]. For the significance test, we separated the French-English human aligned data into 10 data sets of 9 sentences and the English-Chinese human aligned data into 10 data sets of 36 sentences and performed a paired t-test on F1 scores.

3.3.2 EBMT performance

Since our goal is to develop a new alignment method to improve the CMU EBMT system's performance, we evaluated the performance of the CMU EBMT system using SPA, IBM Model 4, and the original internal aligner of the system.

Data For our EBMT experiments we used a subset of the IBM Hansard corpus available from the Linguistic Data Consortium. This corpus is divided into files of 10,000 sentence pairs (with an occasional garbled or missing line which was removed prior to our use), of which we used only files 000 through 099.

The training data consisted of the first 20,000 sentence pairs – files 000 and 001 – for EBMT and the first 700,000 English sentences for the language model. The development test ("Dev") set used for parameter tuning consisted of the first 100 sentences of file 040 and the evaluation test ("Unseen2") set consisted of ten segments of 100 sentences drawn from files 060 and 080. Segmenting the evaluation test set in this manner allowed us to perform Student's t-test as a statistical significance test. Another test ("Unseen1") set consists of 100 source sentences and 200 reference sentences. To see whether the performance is consistent we asked a person to make another reference set for the 100 source sentences such that each source sentence has two reference sentences. The original 100 sentence pairs are mostly drawn from file 060.

[4] We used GIZA++ Och and Ney (2000) for IBM Model 4 alignment.

[5] The score threshold was found empirically by measuring F1 on the hand aligned set.

Evaluation methodology To minimize the initial investment of effort for the EBMT evaluation, we performed a partial exploitation of the SPA and EBMT modules rather than fully incorporating SPA into the EBMT engine [6]. In this partial integration, SPA is used to annotate the training corpus with alignments (both phrasal and word-to-word), and the annotations in the corpus override the EBMT engine's internal aligner. Phrasal alignments are stored as-is, and whenever a partial match against the corpus is exactly equal to the source half of such an alignment, the target half is output as the candidate translation. The word-to-word alignments are used to build a correspondence table (overriding the one which would have been built in the absence of alignment annotations) and that table is consulted as usual to perform alignments of matches for which there is no phrasal alignment from SPA available.

This yields the following training regimens for the alignment methods. To test the old algorithm, we

1. built an EBMT dictionary from the corpus; and
2. indexed the training text using that dictionary.

To test performance with IBM Model 4 alignments, we

1. trained GIZA++ Och and Ney (2000) on the training text;
2. annotated the training corpus with phrasal alignment information using Model 4; and
3. indexed the annotated corpus.

To test performance with SPA, we

1. used GIZA++ to build a dictionary from the training text;
2. ran the SPA aligner on the training text using that dictionary; and

[6] They are fully integrated in Chapter 4

3. indexed the phrasal alignment annotated corpus generated by SPA.

The differently-trained translation systems are then each evaluated on the test set using the BLEU (Papineni et al., 2002) which is the most widely used automatic evaluation metric.

3.4 Results and analysis

3.4.1 Alignment evaluation

Test/Answer	Recall	Prec.	F_1	Len(Test)/Len(Answer)
A/B	0.8588	0.9809	0.9158	0.8755
A/C	0.7427	0.9829	0.8461	0.7556
B/A	0.8968	0.9765	0.9350	0.9184
B/C	0.7834	0.9877	0.8737	0.7931
C/A	0.9590	0.9508	0.9549	1.0086
C/B	0.9686	0.9615	0.9650	1.0074

Table 3.1: Human answer evaluation

As we already mentioned, given a set of parallel sentences, human alignments are not unique. This problem is related to how accurate our evaluations results are. To roughly estimate their accuracy, we used our evaluation metrics to evaluate the human answers by regarding them as machine answers and the machine answers as human answers for the same data set. Table 3.1 shows the human answer evaluation results. In these tests, F_1 varies from 0.8461 to 0.9650. This may give us a rough idea about what score we can aim to achieve. Of course, approaching those values does not mean that the automated aligners are as good as human ones because the errors by the automated aligners might be linguistically serious while human errors are not. It also shows the average target phrase length ratio for the same source phrases in the column of $Len(Test)/Len(Answer)$ [7].

[7] In English/Chinese hand aligned corpus, A, B and C have about 31%, 28% and 3% of unaligned target

Key	Description
random	Random results
positional	Results in proportional positions
oracle	The best possible contiguous results from human answer
SPA-single	SPA - unidirectional alignment (source to target)
SPA-basic	SPA - basic bi-directional alignment
SPA-anchor	SPA - basic + anchor bonus
SPA-len	SPA - basic + length penalty
SPA-dist	SPA - basic + distance penalty
SPA-x_1-x_2..	x_n can be substituted with a,l,d and u. **a**: anchor bonus, **l**: length penalty, **d**: distance penalty, **u**: untranslated word penalty
IBM4-cont	IBM4 - considers the words between the smallest and the largest as the contiguous answer
IBM4-cont-oracle	IBM4 - the best possible contiguous results
IBM4	IBM4 - non-contiguous results
COMB	combined results of the best SPA and IBM4

Table 3.2: Key to the following alignment evaluation tables

For comparing the alignment accuracy, we chose the positional alignment as the base line – as this is the best we can do without any information about the words at all – and the oracle alignment as the goal. Tables 3.3 through 3.6 show the oracle result obtained by each alignment method.

Table 3.3 and Table 3.5 show the best performance by each aligner and Table 3.4 and Table 3.6 show the possibility of improvement for SPA aligners. In Table 3.4 and Table 3.6,

words respectively. And in French-English hand aligned corpus, there are about 5% of unaligned target words.

Aligner	Recall	Prec.	F_1	Len(M)/Len(H)
random	0.3220	0.3722	0.3453	0.8651
positional	0.5823	0.5762	0.5792	1.0105
oracle	0.9056	0.8614	0.8830	1.0513
SPA-single	**0.9426**	0.3560	0.5168	2.6480
SPA-basic	0.8699	0.4739	0.6135	1.8357
SPA-anchor	0.7924	0.4722	0.5918	1.6780
SPA-len(7)	0.7867	0.6104	0.6874	1.2889
SPA-dist(10)	0.8779	0.4673	0.6100	1.8784
SPA-l-u	0.7335	0.6939	*0.7131*	1.0571
SPA-a-l	0.7146	0.5694	0.6338	1.2551
SPA-a-d	0.7981	0.4720	0.5932	1.6910
SPA-l-d	0.7881	0.6036	0.6836	1.3058
SPA-l-d-u	0.7350	0.6841	0.7086	1.0744
SPA-a-l-d	0.7183	0.5687	0.6348	1.2632
SPA-a-l-d-u	0.7034	0.5985	0.6467	1.1754
IBM4-cont	0.8167	0.6043	0.6946	1.3516
IBM4-cont-oracle	0.7271	0.7003	0.7134	1.0383
IBM4	0.7390	**0.8075**	0.7717	0.9152
COMB	0.7563	0.8042	**0.7795**	0.9405

Table 3.3: English-Chinese: Best alignment results evaluation

we reported the best of the top 10 results of the SPA. This shows how closely we pulled the best results toward the top.

Of note, the experiments support the hypothesis that a symmetric method performs better than a unidirectional method: SPA-basic outperformed SPA-single in both Table 3.3 and Table 3.5. Note that the recall of SPA-single is the highest because there is not a length restriction on the target phrases. According to the formula of SPA-single, all the target phrases that include all the maximum probability word translations have the same

Aligner	Recall	Prec.	F_1	Len(M)/Len(H)
SPA-single	**0.9865**	0.4739	0.6402	2.0817
SPA-basic	0.9405	0.6201	0.7474	1.5167
SPA-anchor	0.8980	0.6747	0.7705	1.3310
SPA-len(7)	0.8889	0.7645	0.8220	1.1627
SPA-dist(10)	0.9473	0.6111	0.7429	1.5501
SPA-l-u	0.8767	0.8112	**0.8426**	1.0807
SPA-a-l	0.8621	0.7723	0.8147	1.1162
SPA-a-d	0.9036	0.6692	0.7687	1.3502
SPA-l-d	0.8889	0.7557	0.8169	1.1763
SPA-a-l-d	0.8614	0.7677	0.8119	1.1221
SPA-a-l-d-u	0.8579	0.7805	0.8174	1.0992
COMB	0.7639	**0.8180**	0.7900	0.9338

Table 3.4: English-Chinese: Top 10 alignment results evaluation

alignment score.

Table 3.3 and Table 3.4 show the performance of SPA on English-Chinese data. Here we observe that only two of the penalties (length and untranslated words) helped individually, and the highest overall score was obtained when those two are applied together. Because English and Chinese sentence structures are very different, distance penalty which assumes the same word orders did not help. However, the length penalty worked as expected because it is rare that a target phrase is much longer or much shorter than its source phrase even in a distant language pair. In this language pair, the untranslated word penalty also helped throwing out irrelevant words from the target phrases. But the anchor context did not help as expected. It is possible that 1-to-1 word correspondence is low for this language pair and automatically learned word translation probability is not very discriminative, which consequently leads to less discriminative anchor context scores [8].

[8] Fossum et al. (2008) reported that noisy alignments are more frequent between function words in Chinese-English pair.

Aligner	Recall	Prec.	F_1	Len(M)/Len(H)
random	0.1939	0.2384	0.2139	0.8136
positional	0.6688	0.7290	0.6976	0.9175
oracle	0.9805	0.9377	0.9586	1.0456
SPA-single	**0.8810**	0.2817	0.4269	3.1276
SPA-basic	0.7078	0.7121	0.7099	0.9940
SPA-anchor	0.7798	0.6722	0.7220	1.1602
SPA-len(4)	0.6994	0.7482	0.7230	0.9348
SPA-dist(4)	0.7707	0.7290	0.7493	1.0572
SPA-a-l	0.7522	0.7750	0.7634	0.9705
SPA-a-d	0.8106	0.7096	0.7567	1.1423
SPA-l-d	0.7521	0.7888	0.7700	0.9535
SPA-a-l-d	0.7831	0.7995	0.7912	0.9795
SPA-a-l-d-u	0.7815	0.8014	*0.7913*	0.9751
IBM4-cont	0.8528	0.8293	0.8409	1.0282
IBM4-cont-oracle	0.8132	0.9146	0.8609	0.8891
IBM4	0.7771	**0.9656**	0.8611	0.8048
COMB	0.7817	0.9607	**0.8620**	0.8137

Table 3.5: French-English: Best alignment results evaluation

In Table 3.3, both IBM4 aligners and SPA aligners outperformed the baseline significantly. We evaluated the IBM4 results in three ways: regarding the whole part between the smallest and the largest positions as a contiguous answer fragment (”IBM4-cont”), regarding its best possible contiguous fragment as a contiguous answer fragment (”IBM4-cont-oracle”) and considering it as it is (”IBM4”). Overall the IBM Model 4 aligner showed the best performance and SPA-l-u approached to IBM4-cont-oracle. This means, for the contiguous alignment, IBM4-cont-oracle and SPA-l-u have almost the same performance. The best SPA in Table 3.4 is better than IBM4 in Table 3.3 which means that it is possible to improve SPA and after improvement, SPA might outperform IBM4.

Aligner	Recall	Prec.	F_1	Len(M)/Len(H)
SPA-single	**0.9680**	0.3460	0.5098	2.7977
SPA-basic	0.9038	0.8209	0.8603	1.1010
SPA-anchor	0.9294	0.8432	0.8842	1.1023
SPA-len(4)	0.8822	0.8754	0.8788	1.0078
SPA-dist(4)	0.9382	0.8338	0.8829	1.1252
SPA-a-l	0.9096	0.9026	0.9061	1.0078
SPA-a-d	0.9432	0.8574	0.8983	1.1000
SPA-l-d	0.9159	0.8882	0.9018	1.0312
SPA-a-l-d	0.9231	0.9045	0.9137	1.0205
SPA-a-l-d-u	0.9229	0.9054	**0.9141**	1.0193
COMB	0.7945	**0.9651**	0.8715	0.8233

Table 3.6: French-English: Top 10 alignment results evaluation

Test-set-id	COMB(en-cn)	IBM4(en-cn)	COMB(fr-en)	IBM4(fr-en)
0	0.6502	0.6223	0.8737	0.8725
1	0.7506	0.7401	0.8696	0.8715
2	0.7413	0.7348	0.8240	0.8182
3	0.7386	0.7332	0.8776	0.8770
4	0.7879	0.7835	0.8995	0.9034
5	0.8363	0.8328	0.8659	0.8635
6	0.7936	0.7869	0.8164	0.8169
7	0.7964	0.7956	0.8201	0.8157
8	0.7686	0.7599	0.8906	0.8953
9	0.8048	0.8012	0.8683	0.8638
P-value	0.01		0.5	

Table 3.7: The significance test for COMB and IBM4

Table 3.5 and Table 3.6 show the performance of SPA on French-English data. Here we observe that each penalty (length, distance, anchor, and untranslated words) helped SPA individually, and that, in fact, the highest score was obtained when all the four penalties were applied together. French and English are very similar languages in their grammatical structures. The length and distance penalties are helpful because for a pair of translation phrases, their lengths are comparable, and their positions are similar in their sentences. The anchor context supported the target phrase by the surrounding word translation score. The untranslated word penalty helps throw out irrelevant words from the target phrases. Interestingly, the "positional" result is very close to "SPA-basic" in F1, and this shows that the two languages are very close in sentence structure. Because the oracle SPA alignment gives higher scores than IBM4 in the top 10 as seen in Tables 3.6 and 3.5, SPA shows the potential for improvement and ,with such, might outperform IBM4.

Table 3.7 shows the results of the combined aligner ("COMB") for both language pairs, and the COMB outperforms IBM. We use e^{-11} and e^{-12} for French-English and English-Chinese thresholds respectively in probability space (Because we use log space instead of probability space for efficient computation in our actual implementation, we have these empirically obtained values as thresholds.). Our significance test shows that for English-Chinese, the combined version significantly outperforms IBM4 while for French-English, the difference is only slightly significant.

3.4.2 EBMT performance

Because we developed SPA to help the EBMT system generate better translation, so we also evaluate its effect on EBMT translation quality.

After tuning several key parameters in the EBMT system separately for each alignment algorithm in use, we obtained the scores shown in Table 3.8.

In Table 3.8, we observe that SPA outperforms EBMT - the old aligner in the CMU EBMT system by a marked difference. For the Dev, Unseen1, and Unseen2 data set, SPA has 35%, 20%, and 28% higher BLEU scores than EBMT, respectively, which is a great improvement.

	Dev	**Unseen1**	**Unseen2**
EBMT	0.1632	0.2400	0.1346
SPA	0.2214	**0.2896**	0.1729
IBM4	0.2197	0.2785	0.1755
COMB	**0.2240**	0.2815	0.1751

Table 3.8: French-English BLEU scores by aligners

We also observe that IBM4 and COMB significantly outperformed the EBMT but the performance differences among SPA, IBM4, and COMB are very small. For Dev, Unseen1, and Unseen2, the winners are different - COMB for Dev, SPA for Unseen1, and IBM4 for Unseen2.

Our significance test shows that SPA, IBM4, and COMB perform significantly better than EBMT, but that the differences among SPA, IBM4, and COMB are not significant ($0.38 \leq p \leq 0.45$).

In this chapter, we have demonstrated that, with properly chosen constraints, SPA shows nice performance in both alignment accuracy and translation. However, for each constraint, we simply combined the feature score one-by-one using linear interpolation. We need a reasonable framework where the weights of constraints are automatically tuned together. Furthermore, we did not use word alignment information, which is a by-product when we build a dictionary using word alignment models [9]. The word alignment information has source/target word mappings per sentence pair, and we think it will be beneficial for SPA, which finds translations from each sentence pair under consideration, because word translation probabilities in the dictionaries are calculated over the whole training corpora. SPA also outputs only contiguous target phrases that may include irrelevant words, but we did not show how it affected both alignment quality and translation quality.

In the next chapter, we investigate a framework in which we combine the constraint feature scores together, use external word alignment information, and employ non-contiguous phrasal alignment to exclude irrelevant target words.

[9] IBM word alignment models in our experiments.

Chapter 4

SPA enhancements

In Chapter 3, we investigated the improvements SPA made over the baseline EBMT system. In spite of these initial improvements, we discussed three points for further improvement at the end of the chapter. First, SPA provides a single final phrasal alignment score to the translation system by using a heuristic-based function for each constraint feature and combining them one-by-one in a simple linear interpolation that might be improved by considering a more sophisticated way of combining feature scores. Secondly, the improvements described in Chapter3 did not use word alignment information, a by-product of building a dictionary using IBM word alignment models. Word translation probabilities in the dictionaries are calculated over the whole training corpus. However, because the word alignment information has source/target word mappings for each sentence pair, we think its use will be beneficial for SPA, which finds translations from each sentence pair under consideration. Finally, the SPA outputs only contiguous target phrases that may include irrelevant target words. We need to investigate the possibility of excluding such irrelevant target words.

In this chapter, we first discuss modifying SPA to return multiple translations with multiple features and to tune weights for the various features via minimum error rate training. Then we consider how to employ external alignment information in SPA. Finally, we investigate non-contiguous alignment.

4.1 Multiple Translations

We made two immediate modifications to SPA. We first changed SPA to return multiple feature values instead of a single combined final value. We next made changes on some feature values. The final SPA feature functions are explained in Section 4.4.3. Note that, in the feature function list, each lex function is a uni-directional SPA score , $bonus$ is an extended version of *Anchor Context* , each $untrans$ is a uni-directional *Untranslated Penalty* , p is a modified version of *Length Penalty*. $penalty$ was newly introduced, and the old feature *Distance Penalty* was removed because it is language specific.

Second, SPA can return top N_{SPA} candidates for a source match where N_{SPA} is the maximum number of alternative translations. Now, in SPA, we are given a source phrase $\mathbf{f} = f_{i+1}^{i+k} = f_{i+1}, ..., f_{i+k}$ and a source/target sentence pair (**F**, **E**). We first set a range $[r_{start}, r_{end}]$ from which we draw potential translation candidates using the position of the source phrase in the source sentence.

$$r_{start} = (i+1) \times \frac{|\mathbf{E}|}{|\mathbf{F}|} \tag{4.1}$$

$$r_{end} = (i+k) \times \frac{|\mathbf{E}|}{|\mathbf{F}|} \tag{4.2}$$

These are then modified by applying a pre-defined window size W [1].

$$r_{start} \leftarrow max(1, r_{start} - W) \tag{4.3}$$

$$r_{end} \leftarrow min(L, r_{end} + W) \tag{4.4}$$

Next all the possible contiguous target fragments from the range defined by r_{start} and r_{end} are assessed as candidate translations of the given source phrase. Now the candidate set C is:

$$C = \{c_k | c_k = e_m, ...e_n, m <= n, m >= r_{start}, n <= r_{end}\} \tag{4.5}$$

Figure 4.1 shows an example of a calculated range with a window size of 2.

[1] We used *W*=3 which is empirically obtained.

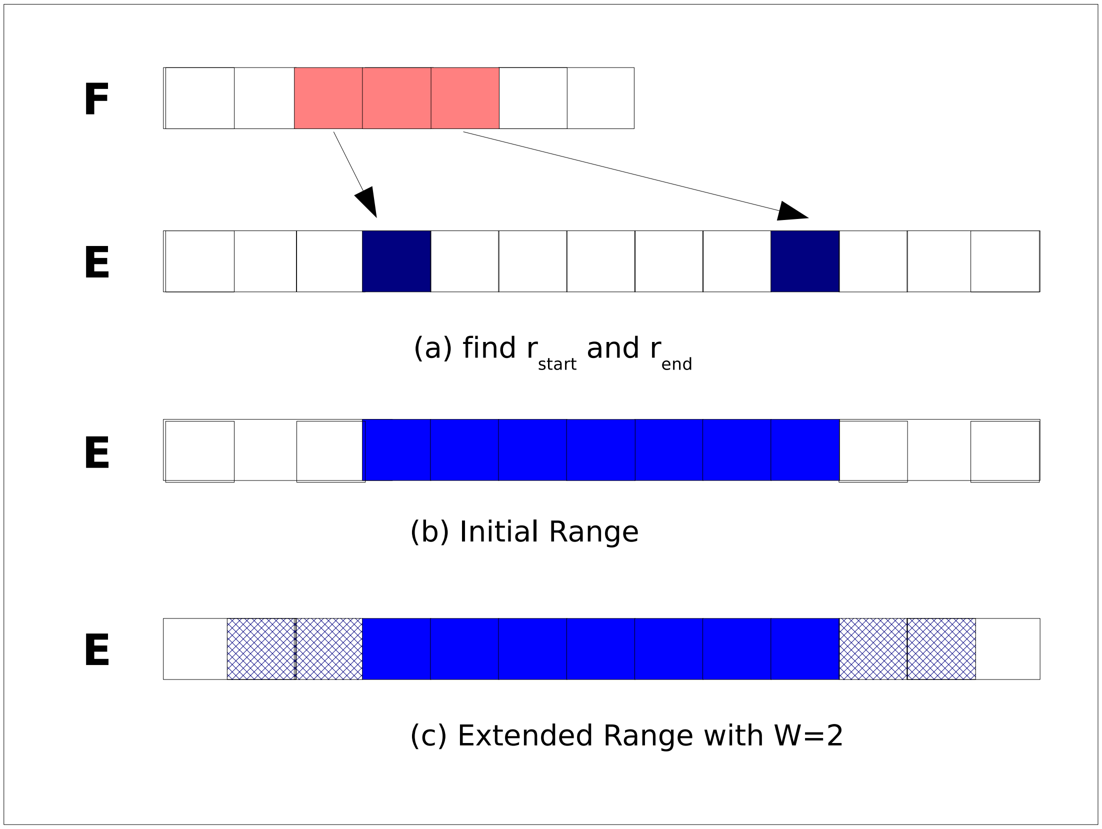


Figure 4.1: Defining a candidate range from external alignments

This approach enables SPA to reduce the search space in finding target candidates. The basic SPA theoretically assesses $\frac{L(L+1)}{2}$ candidates when the target sentence is L words long. However this approach allows SPA to assess at most $\frac{L'(L'+1)}{2}$ candidates where $L' = r_{start} - r_{end} + 1$. These candidates are then filtered before score calculation so that they are not too much longer or shorter than the source phrase. To remove unrealistically long or short target candidates, we apply a predefined length ratio to the source phrase length and filter out those that are not within the calculated range of length.

$$\begin{aligned} C &= \{\mathbf{c}_k | \mathbf{c}_k = e_m, ...e_n, m <= n, m >= r_{start}, n <= r_{end}, \\ &\qquad n - m + 1 <= |\mathbf{f}| * R_{max}, n - m + 1 >= |\mathbf{f}| * R_{min}\} \end{aligned} \tag{4.6}$$

where $\mathbf{f}$ denotes the source phrase and R_{min} and R_{max} denote the acceptable maximum ratio and minimum ratio respectively [2].

For the considered candidates in C, SPA calculates their feature values. Then it multiplies two lex values and $p(|\mathbf{e}|)$ value to get an *alignment score* that will be used in obtaining C_{sorted} which is a sorted set of C.

$$C_{sorted} = \{\mathbf{c}_p | 0 \leq p < |C|, AS(\mathbf{c}_{p-1}) \geq AS(\mathbf{c}_p) \; where \; 0 < p < |C|\} \tag{4.7}$$

where $AS(\mathbf{c}_p)$ is the *alignment score* of the candidate $\mathbf{c}_p$. Finally it returns top N_{SPA} candidates whose alignment score satisfies a score ratio criterion:

$$C_{N_{SPA}} = \{\mathbf{c}_p | AS(\mathbf{c}_p) \geq Ratio_{SPA} \times AS(\mathbf{c}_0), 0 \leq p < min(N_{SPA}, |C_{sorted}|)\} \tag{4.8}$$

where $Ratio_{SPA}$ is a ratio value between 0.0 and 1.0. The N_{SPA} and $Ratio_{SPA}$ are configurable parameters in the EBMT system.

The $AS(\mathbf{c}_p)$ is passed to the EBMT system along with the SPA feature scores. The EBMT engine uses $AS(\mathbf{c}_p)$ when it prunes candidate translations to have only N_{EBMT} translation candidates for each source match. The N_{EBMT} is a configurable parameter in the EBMT system to specify the maximum number of translation alternatives for each source match. The $AS(\mathbf{c}_p)$ is also used in decoding like other SPA feature scores.

4.2 Framework for parameter tuning

Decoder

Then we modified our decoder to work in a log-linear model. For a source sentence $\mathbf{F} = f_1^J = f_1, ..., f_j, ..., f_J$ and a possible target sentence $\mathbf{E} = e_1^I = e_1, ..., e_i, ..., e_I$,

$$\begin{aligned} Pr(\mathbf{E}|\mathbf{F}) &= p_{\lambda_1^M}(\mathbf{E}|\mathbf{F}) & (4.9) \\ &= \frac{exp[\sum_{m=1}^{M} \lambda_m h_m(\mathbf{E}, \mathbf{F})]}{\sum_{e'^I_i} exp[\sum_{m=1}^{M} \lambda_m h_m(e'^I_1, \mathbf{F})]} & (4.10) \end{aligned}$$

[2] We used 3 and 5 for R_{min} and R_{max} respectively in our experiments. They were empirically chosen and worked reasonably well.

where we have a set of M feature functions $h_m(e, f), m = 1, ..., M$ and for each feature function, there exists a model parameter $\lambda_m, m = 1, ..., M$.

The decoder uses the feature scores calculated by the EBMT engine itself and the feature scores returned by SPA. The EBMT feature scores are described in Section 2.1.2 and the SPA feature scores consist of the alignment score $AS(\mathbf{e})$ and the feature scores described in Section 4.4.3.

Parameter Tuning

To optimize the parameter set, we use the Minimum Error Rate Training (MERT) approach described by Och (2003) using $BLEU$ as the error criterion. Zhao and Waibel (2005) used MERT to extract translation phrases independently of a decoder, but we use it in our decoder so that the feature weights are optimized directly for translation quality. The approach assumes that the number of errors for a set of sentences $\mathbf{E}_1^S$ is obtained by summing the errors for the individual sentences: $E(\mathbf{R}_1^S, \mathbf{E}_1^S) = \sum_{s=1}^{S} E(\mathbf{R}_s, \mathbf{E}_s)$. The goal is to obtain a minimal error count for a representative corpus $\mathbf{F}_1^S$ with given reference translations $\hat{\mathbf{E}}_1^S$ and a set of K different candidate translations $C_s = \mathbf{E}_{s,1}, ..., \mathbf{E}_{s,K}$ for each input sentence $\mathbf{F}_s$.

$$\hat{\lambda}_1^M = argmin_{\lambda_1^M} \sum_{s=1}^{S} E(\mathbf{R}_s, \hat{\mathbf{E}}(\mathbf{F}_s; \lambda_1^M)) \tag{4.11}$$

$$= argmin_{\lambda_1^M} \sum_{s=1}^{S} \sum_{k=1}^{K} E(\mathbf{R}_s, \mathbf{E}_{s,k}) \delta(\hat{\mathbf{E}}(\mathbf{F}_s; \lambda_1^M), \mathbf{E}_{s,k}) \tag{4.12}$$

with

$$\hat{\mathbf{E}}(\mathbf{F}_s; \lambda_1^M) = argmax_{\mathbf{E} \in C_s} \sum_{m=1}^{M} \lambda_m h_m(\mathbf{E}|\mathbf{F}_s) \tag{4.13}$$

In this work, we use the minimum error rate implementation in the Moses system by Koehn et al. (2007).

Figure 4.2 shows the MERT integration into the EBMT system. Once the EBMT system builds a dynamic phrase table which is basically a lattice, it stores the table in a file. The rest of the MERT process is almost the same as the MERT process in the Moses system.

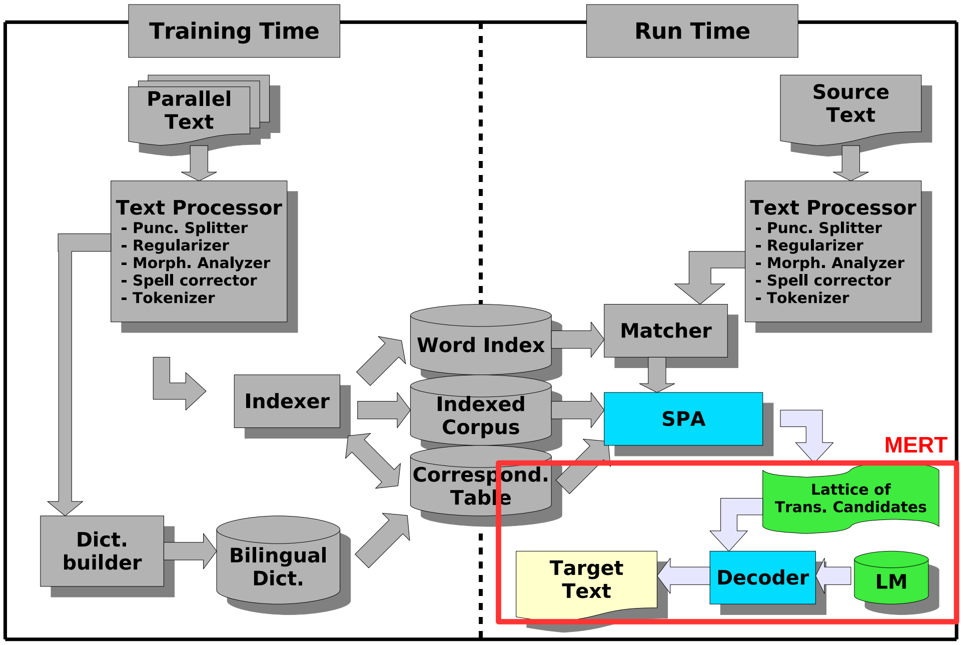


Figure 4.2: Modified components for parameter tuning

1. The EBMT decoder [3] loads parameter weights (feature weights) and the table.

2. It finds top-N hypothesis translations.

3. MERT finds new optimized weights for feature scores.

4. The EBMT returns to step 1 if the weights did not converge.

The tuned feature weights are used in future translations.

[3] The EBMT decoder can run as a stand-alone version taking a lattice as input.

4.3 Exploiting external word alignment

In Chapter 3, we assumed that we have only bilingual probabilistic translation dictionaries. In general, these dictionaries are trained in two ways. One group calculates translation likelihood from the number of co-occurrences of word pairs in a comparable corpus. They count the occurrences of all the possible source and target word pairs in each possible translation segments. They then accumulate these statistics through the entire corpus and calculate translation likelihood. For example, when they define translation likelihood conditional probability, they calculate it as:

$$p(e_j|f_i) = \frac{count(e_i, f_j)}{\sum_k count(f_i, e_k)} \tag{4.14}$$

The other group learns word alignment, which is a set of word links [4], or translated word pairs in a parallel corpus. They then calculate conditional probability based on the word links from the alignment.

$$p(e_j|f_i) = \frac{link\_count(f_i, e_j)}{\sum_k link\_count(f_i, e_k)} \tag{4.15}$$

Among the statistical alignment methods, IBM Model 4 is close to the state-of-the-art aligners. Although some researchers using other alignment methods have reported improvements over the IBM model, the improvements are not large, and often they use IBM Model 4 alignments as an important factor as in Taskar et al. (2005)'s work.

For the research discussed in Chapter 3, we had trained dictionaries using IBM Model 4, hence we had high quality word alignment information which we did not use. In this chapter, we investigate how to use external word alignment information in SPA. We first use the external alignment information for our contiguous SPA and then investigate non-contiguous alignment in the next section.

For contiguous SPA, we use external alignment information to find a range from which we draw target candidates. This is helpful when the proportional range does not include real alignments. Now, in SPA, we are given a source phrase, a source/target sentence pair,

[4] We use word links and mappings interchangeably hereafter in this thesis.

and word alignments for the pair. From the target words that are aligned from the source phrase words, we first find the ones with the smallest index and the largest index and set r_{start} and r_{end} with them respectively to define a range $[r_{start}, r_{end}]$ from which we draw possible target candidates. The rest of process is the same as the modified contiguous SPA in Section 4.1. Figure 4.1 shows an example of a calculated range with a window size of 2.

4.4 Non-contiguous alignment

Given a pair of translation sentences and a source phrase, we find a target translation phrase that consists of aligned target words from the source words in the source phrase. When all the target words between the first aligned target word and the last aligned target word are aligned from any of the source words in the source phrase, we say that the target phrase is *contiguous*. But when there is any unaligned target word, we say that the target phrase is *non-contiguous* and call a series of consecutive unaligned target words a *gap*.

For the hand-aligned data we used in Chapter 3, we counted how many times the 3-8 words long source phrases are aligned to non-contiguous target phrases. Depending on whether we count an unaligned target word by human as a part of a gap (case 1) or not (case 2), the portions of non-contiguous alignments are different. The statistics are reported in Table 4.1. For English-Chinese, 41.7% to 63.8% of source phrases are aligned to non-contiguous target phrases and for French-English, 9.1% to 29.6% of source phrases are aligned to non-contiguous target phrases. These portions are significant and led us to study non-contiguous SPA. Another observation is that the close language pair of French-English has less word order discrepancy which leads to less non-contiguity than the distance language pair of English-Chinese, and we may benefit more by using non-contiguous SPA for the distant language pair.

In this section, we describe our basic idea of scoring. Simply, given a source phrase and a target phrase with a single gap, we can calculate the alignment score for them by boosting the alignment score when the gap and the outside of the source fragment have a translation

Language Pair	English-Chinese	French-English
Number of sentence pairs	386	91
Number of source phrases	27,286	12,446
Number of non-contiguous target phrases (case 1)	63.8%	41.7%
Number of non-contiguous target phrases (case 2)	29.6%	9.1%

Table 4.1: Non-contiguous alignment statistics on the hand-aligned corpora

relationship or the outside of the candidate target fragment and the outside of the source fragment have a translation relationship. On the other hand, we penalize the score when the candidate target fragment and the outside of the source fragment have a translation relationship, when the gap and the source fragment have a translation relationship, or when the outside of the candidate target fragment and the source fragment have a translation relationship.

Figure 4.3 shows the boosting area and the penalizing area. Areas 3 and 6 represent non-contiguous alignment for the source and target fragments, areas 1, 2, 4, 5, 7 and 8 are the boosting areas, and areas A, B, C, D, E, F and G are the areas that are penalized.

One example of the formula for the alignment score could be written as follows:

$$\begin{aligned} Score_{\mathbf{e}} \leftarrow \quad & \alpha \times ScoreF(i,i) \\ & -\beta \times ScoreF(i,g) \\ & -\gamma \times ScoreF(i,o) \\ & +\delta \times ScoreF(o,o) \\ & +\epsilon \times ScoreF(o,g) \\ & -\zeta \times ScoreF(o,i) \end{aligned} \tag{4.16}$$

where

$$ScoreF(i,g) = P(i \leftrightarrow g) \tag{4.17}$$

given a target fragment $\mathbf{e}$ and α, β, γ, δ, ϵ, and ζ are all positive. Here the first argument of $ScoreF()$ is an area in the source sentence, and the second argument is an area in

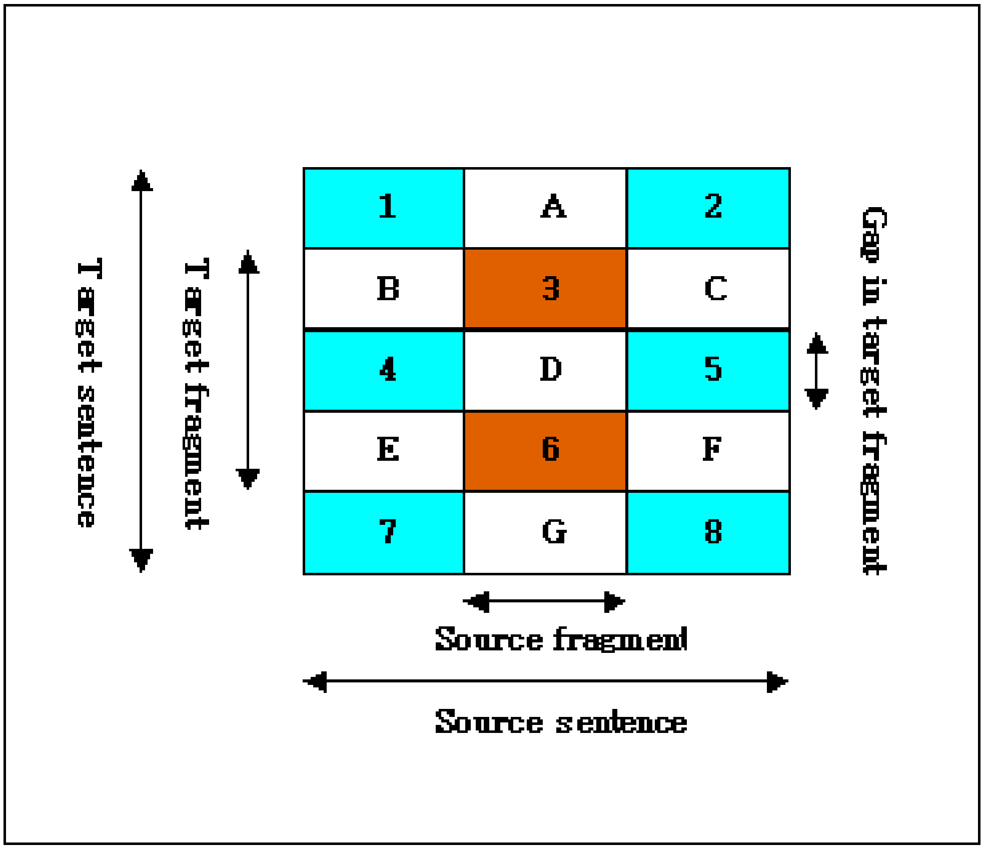


Figure 4.3: Non-contiguous alignment

the target sentence. The area labels i, o, and g represent *inside of the fragment*, *outside of the fragment in the sentence*, and *the gap in the fragment* respectively. Therefore, in Figure 4.3, $ScoreF(i, i)$ is the score for area 3 and 6, $ScoreF(i, g)$ is the score for area D, $ScoreF(i, o)$ is the score for area A and G, $ScoreF(o, o)$ is the score for area 1, 2, 7 and 8, $ScoreF(o, g)$ is the score for area 4 and 5, $ScoreF(o, i)$ is the score for area B, C, E and F. For example, given a candidate target fragment $\mathbf{e}$, $ScoreF(i, g)$ function calculates the alignment score between the source fragment and the gap in the target fragment.

4.4.1 A computationally feasible approach

To calculate the score for a target phrase with multiple gaps, we can extend the scoring approach described in the previous section. We calculate the inside score and outside score and use them to boost or penalize the score.

However, identifying the gaps in an efficient way is a challenging task. Given a source phrase in basic SPA, we had

$$\frac{L \times (L+1)}{2} \tag{4.18}$$

target candidates where L is the target sentence length. In the non-contiguous alignment, we have

$$2^L \tag{4.19}$$

possible candidates because each target word can be included or excluded from a candidate. Because Moses implements an accurate and widely-used symmetric word alignment method, the *Refined Method*, and the alignment is inherently non-contiguous, we investigate exploiting it for our non-contiguous alignment base.

First, given a source phrase, we start from all of the target words that are aligned to any of the source phrase words in the Moses word alignment. Secondly, we check all the target words near the boundaries within a window size of W [5]. If a target word is already aligned and its outside score is larger than the inside score, the word could potentially be removed and we say it is *removable*. If a target word is not aligned and its inside score is larger than the outside score, the word could potentially be included and we say it is *includable*. Thirdly, by excluding or including each removable/includable word, we generate target candidates. So, if we have i includable words and r removable words, we then have

$$2^{i+r} \tag{4.20}$$

target candidates. For $i+r <= L$, $2^{i+r} <= 2^L$, but in reality $i+r << L$ and $2^{i+r} <<< 2^L$.

Figure 4.4 illustrates a non-contiguous alignment procedure using Moses alignment as its basis. Solid black boxes denote non-contiguous external word alignment, or Moses

[5] We used $W = 1$ in our experiments assuming that target words are close to each other.

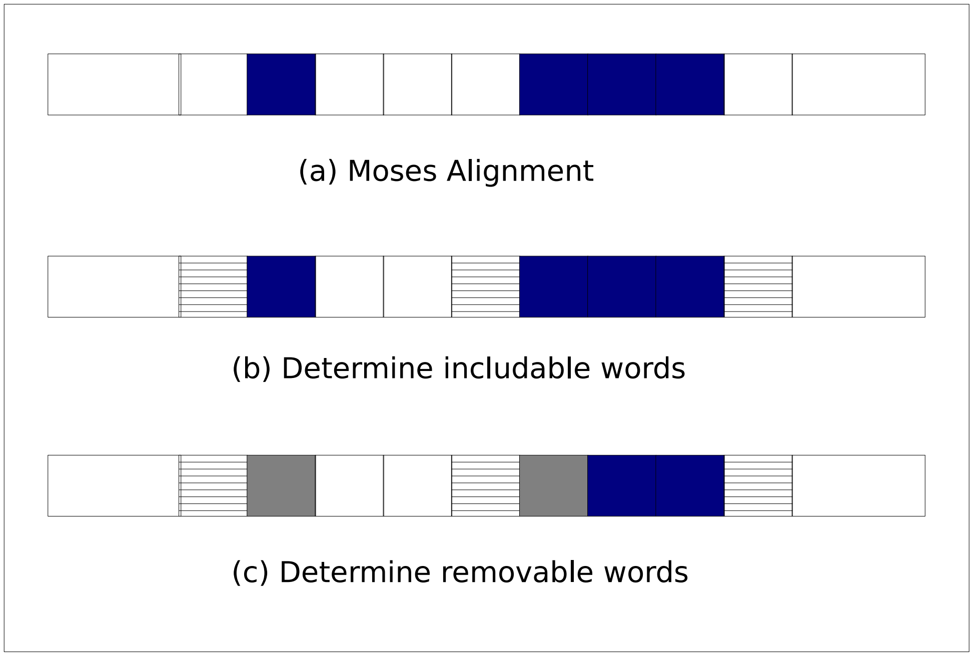


Figure 4.4: Non-contiguous alignment extended from Moses alignment with W=1: (a) Moses alignment, (b) includable words are determined, (c) removable words are determined.

word alignment output in this case. The striped boxes denote includable words, and the gray boxes denote removable words. Because we have three striped boxes and two gray boxes, we generate $2^{3+2} = 2^5 = 32$ candidate alignments in this case.

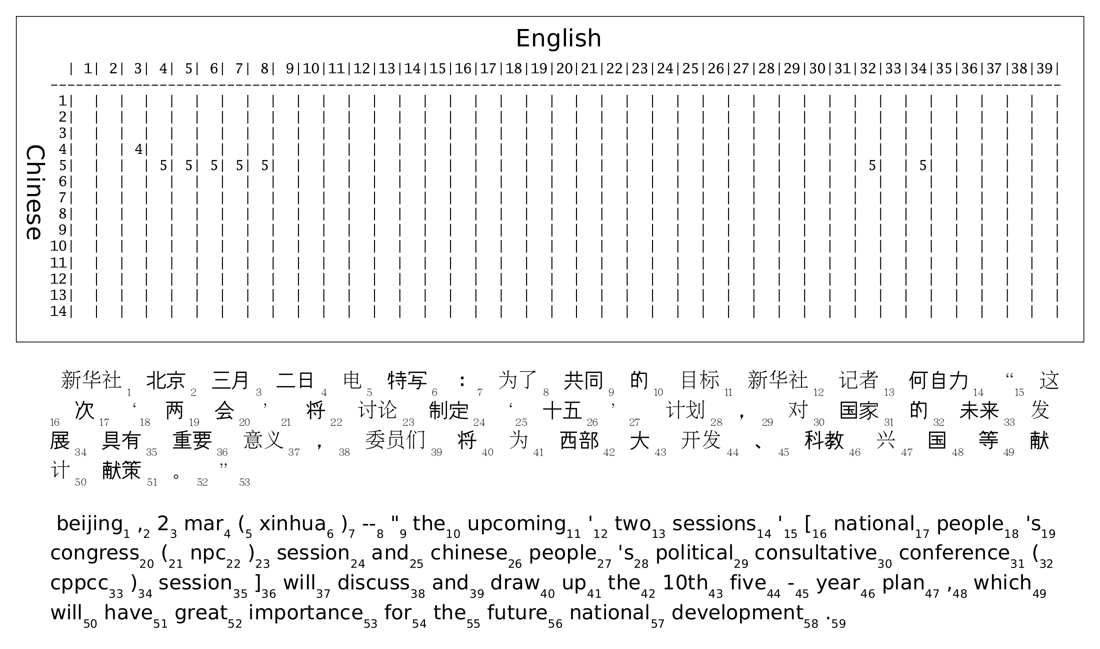
We then calculate feature scores, including supporting feature scores and penalizing feature scores for each candidate, and pass them to the decoder so that they are used in the translation with their tuned parameters in a separate step.

In this thesis work, we used the decoder in the EBMT system which did not remove gaps in non-contiguous alignments by interlocking target phrases. Instead, it simply dropped gaps from non-contiguous alignments if it gives a better LM score or discarded non-contiguous alignments.

4.4.2 Pre-processing for unsupervised external alignment

In general, unsupervised alignment is particularly difficult for linguistically distant language pairs. For example, some languages are SVO and some are SOV [6]. Some have determiners and some do not [7]. Some have detailed case markers and some do not [8]. Even if we have a fairly good amount of data, we have a lot of incorrect word alignments due to such linguistic differences. Finding these incorrect alignments is as difficult as finding correct alignments.

We sometimes observe that our external aligner finds very isolated words (out-liers) as alignments and that most of them are incorrect alignments.



```
                                            English
   | 1| 2| 3| 4| 5| 6| 7| 8| 9|10|11|12|13|14|15|16|17|18|19|20|21|22|23|24|25|26|27|28|29|30|31|32|33|34|35|36|37|38|39|
-----------------------------------------------------------------------------------------------------------------------
  1|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
  2|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
  3|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
  4|  |  | 4|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
  5|  |  |  | 5| 5| 5| 5| 5|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 5|  | 5|  |  |  |  |  |
  6|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
  7|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
  8|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
  9|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
 10|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
 11|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
 12|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
 13|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
 14|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
```

新华社$_{1}$ 北京$_{2}$ 三月$_{3}$ 二日$_{4}$ 电$_{5}$ 特写$_{6}$ ：$_{7}$ 为了$_{8}$ 共同$_{9}$ 的$_{10}$ 目标$_{11}$ 新华社$_{12}$ 记者$_{13}$ 何自力$_{14}$ “$_{15}$ 这
$_{16}$ 次$_{17}$ ‘$_{18}$ 两$_{19}$ 会$_{20}$ ’$_{21}$ 将$_{22}$ 讨论$_{23}$ 制定$_{24}$ ‘$_{25}$ 十五$_{26}$ ’$_{27}$ 计划$_{28}$ ，$_{29}$ 对$_{30}$ 国家$_{31}$ 的$_{32}$ 未来$_{33}$ 发
展$_{34}$ 具有$_{35}$ 重要$_{36}$ 意义$_{37}$ ，$_{38}$ 委员们$_{39}$ 将$_{40}$ 为$_{41}$ 西部$_{42}$ 大$_{43}$ 开发$_{44}$ 、$_{45}$ 科教$_{46}$ 兴$_{47}$ 国$_{48}$ 等$_{49}$ 献
计$_{50}$ 献策$_{51}$ 。$_{52}$ ”$_{53}$

beijing$_{1}$,$_{2}$ 2$_{3}$ mar$_{4}$ ($_{5}$ xinhua$_{6}$)$_{7}$ --$_{8}$ "$_{9}$ the$_{10}$ upcoming$_{11}$ '$_{12}$ two$_{13}$ sessions$_{14}$ '$_{15}$ [$_{16}$ national$_{17}$ people$_{18}$'s$_{19}$ congress$_{20}$ ($_{21}$ npc$_{22}$)$_{23}$ session$_{24}$ and$_{25}$ chinese$_{26}$ people$_{27}$'s$_{28}$ political$_{29}$ consultative$_{30}$ conference$_{31}$ ($_{32}$ cppcc$_{33}$)$_{34}$ session$_{35}$]$_{36}$ will$_{37}$ discuss$_{38}$ and$_{39}$ draw$_{40}$ up$_{41}$ the$_{42}$ 10th$_{43}$ five$_{44}$ -$_{45}$ year$_{46}$ plan$_{47}$,$_{48}$ which$_{49}$ will$_{50}$ have$_{51}$ great$_{52}$ importance$_{53}$ for$_{54}$ the$_{55}$ future$_{56}$ national$_{57}$ development$_{58}$.$_{59}$

Figure 4.5: Removing isolated alignment links

Figure 4.5 shows an example of an unlikely isolated alignment for a Chinese-English

[6] SVO languages include English, French and Chinese and SOV include Korean, Japanese and Turkish.

[7] For example, English determiner ’the’ does not exist in Korean.

[8] For example, Korean has subject case markers but English does not.

sentence pair. The latter parentheses far away from the majority of aligned target words are incorrect. We want to remove those isolated word alignments that are potentially erroneous. Given a source phrase $f_i, ..., f_{i+m}$ and a non-contiguous target phrase $e_j, ..., e_{j+n}$, we can improve the alignment by removing the out-lier alignments as follows:

1. We collect all the contiguous target fragments $C_0 = \{\mathbf{c}_1, ..., \mathbf{c}_{k_0}\}$ and calculate their average inside score.

2. From the center of source phrase positions $sp = (\sum_{p=i}^{i+m} p)/(m+1)$, we calculate an expected target position $tp_{expected} = sp \times \frac{|\mathbf{E}|}{|\mathbf{F}|}$. In this approach, we assume that the source and target sentences have the same word order.

3. If the target fragments are scattered within a range which is R times the length of the fragments, we stop. Formally, $last\_word(\mathbf{c}_{k_t}) - first\_word(\mathbf{c}_1) + 1 <= R \times \sum_{\mathbf{c}_m \in C_t} |\mathbf{c}_m|$ where $R >= 1$, stop [9]

4. If the first fragment score is less than that of the last one, we remove the first fragment. Formally, $score(\mathbf{c}_1) < score(\mathbf{c}_{k_t})$, then $C_{t+1} \leftarrow C_t - \{\mathbf{c}_1\}$ and go to step 3.

5. If the last fragment score is less than that of the first one, we remove the last fragment. Formally, $score(\mathbf{c}_1) > score(\mathbf{c}_{k_t})$, then $C_{t+1} \leftarrow C_t - \{\mathbf{c}_{k_t}\}$ and go to step 3.

6. If $score(\mathbf{c}_1) == score(\mathbf{c}_{k_t})$, we remove the more distant fragment from $tp_{expected}$ in C_t to get C_{t+1} and go to step 3 (Remove a random one if their distances from $tp_{expected}$ are the same).

The $score$ function we used in this approach calculates average word translation probability.

$$score(\mathbf{c}_k) = (\prod_{e \in \mathbf{c}_k} max(max_{j=i}^{i+m} p(e|f_j), \epsilon))^{\frac{1}{|\mathbf{c}_k|}} \tag{4.21}$$

We also considered filling gaps in non-contiguous alignment. We wanted to fill gaps that have reasonably good inside scores compared to their respective outside scores. But in Moses alignment, which we used as external alignment source, there were so few gaps which satisfied our criteria that we did not consider it any more.

[9] We empirically obtained $R = 4$.

4.4.3 Alignment score features

For each target candidate translation for a given source phrase, we calculate multiple feature scores. To help find the best possible translation, these feature scores are then combined for use in decoding.

We denote the given source phrase by $\mathbf{f}$ and outside word sequences in the source sentence by $\mathbf{f}^c$. Likewise, we use $\mathbf{e}$ and $\mathbf{e}^c$ for the current target candidate phrase and outside the target phrase in the target sentence respectively. According to our setting, $\mathbf{f}$ is contiguous and $\mathbf{f}^c$, $\mathbf{e}$, and $\mathbf{e}^c$ can be either contiguous or non-contiguous.

- $lex(\mathbf{e}|\mathbf{f})$

  This feature holds source-to-target lexical translation evidence. A very small probability value ϵ was used to avoid zero production. We used one tenth of the smallest translation probability value in the dictionary as ϵ.

  $$lex(\mathbf{e}|\mathbf{f}) = (\prod_{e_i \in \mathbf{e}} max(max_{f_j \in \mathbf{f}} tr(e_i|f_j), \epsilon))^{|\mathbf{e}|} \tag{4.22}$$

- $lex(\mathbf{f}|\mathbf{e})$

  This feature holds target-to-source lexical translation evidence.

  $$lex(\mathbf{f}|\mathbf{e}) = (\prod_{f_i \in \mathbf{f}} max(max_{e_j \in \mathbf{e}} tr(f_i|e_j), \epsilon))^{|\mathbf{f}|} \tag{4.23}$$

- $bonus(\mathbf{f}, \mathbf{e})$

  This feature holds lexical translation evidence of the outsides of the source and target phrases.

  $$\begin{aligned} bonus(\mathbf{f}, \mathbf{e}) \;\; = \;\; & (\prod_{e_i \in \mathbf{e}^c} max(max_{f_j \in \mathbf{f}^c} tr(e_i|f_j), \epsilon))^{|\mathbf{e}^c|} \\ & \times(\prod_{f_i \in \mathbf{f}^c} max(max_{e_j \in \mathbf{e}^c} tr(f_i|e_j), \epsilon))^{|\mathbf{f}^c|} \end{aligned} \tag{4.24}$$

- $penalty(\mathbf{f}, \mathbf{e})$

  This feature holds lexical translation evidence that shows the current pair is not a good translation pair. The lexical translation score is calculated between $inside$ and $outside$ phrases.

$$\begin{aligned} penalty(\mathbf{f}, \mathbf{e}) \quad = \quad & (\prod_{e_i \in \mathbf{e}^c} max(max_{f_j \in \mathbf{f}} tr(e_i|f_j), \epsilon))^{|\mathbf{e}^c|} \\ & \times (\prod_{f_i \in \mathbf{f}^c} max(max_{e_j \in \mathbf{e}} tr(f_i|e_j), \epsilon))^{|\mathbf{f}^c|} \\ & \times (\prod_{e_i \in \mathbf{e}} max(max_{f_j \in \mathbf{f}^c} tr(e_i|f_j), \epsilon))^{|\mathbf{e}|} \\ & \times (\prod_{f_i \in \mathbf{f}} max(max_{e_j \in \mathbf{e}^c} tr(f_i|e_j), \epsilon))^{|\mathbf{f}|} \end{aligned} \tag{4.25}$$

- $untrans(\mathbf{f})$

  This feature counts the number of source words in **f** that are not translation words from the target words in **e**.

  $$untrans(\mathbf{f}) = \sum_{f_i \in \mathbf{f}} f(f_i, \mathbf{e}) \tag{4.26}$$

  where $f(f_i, \mathbf{e})$ is 1 if $max_{e_j \in \mathbf{e}} tr(f_i|e_j) == \epsilon$, 0 otherwise.

- $untrans(\mathbf{e})$

  This feature counts the number of target words in **e** that are not translation words from the source words in **f**.

  $$untrans(\mathbf{e}) = \sum_{e_i \in \mathbf{e}} f(e_i, \mathbf{f}) \tag{4.27}$$

  where $f(e_i, \mathbf{f})$ is 1 if $max_{f_j \in \mathbf{f}} tr(e_i|f_j) == \epsilon$, 0 otherwise.

- $p(|\mathbf{e}|)$

  Given a source phrase **f**, we assume that the length of its translation is a Gaussian distribution with $\mu = |\mathbf{f}|$ and $\sigma = 1$.

  $$p(|\mathbf{e}|) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(|\mathbf{e}|-|\mathbf{f}|)^2}{2\sigma^2}} \tag{4.28}$$

  $\sigma = 1$ was chosen empirically.

4.5 Evaluation

In this evaluation we measured translation performance differences among SPA variants. The phrasal aligners we compared are:

- **cSPA-mX**
  This is the modified **c**ontiguous SPA described in Section 4.1. It returns X **m**ultiple translation alternatives drawn from a proportionally determined target range.

- **cSPA-amX**
  This is the same as *cSPA-mX* except that this uses external word **a**lignment information to find a target range for translation candidates. This aligner is described in Section 4.3.

- **cSPA-A**
  This is an SPA aligner that returns a single contiguous target phrase that spans from the first aligned target word position to the last aligned target word position. The positions are obtained from the external alignment information.

- **cSPA-AmX**
  This SPA aligner returns multiple target alignment candidates with the contiguous external alignment phrase at the top. The multiple candidates are obtained from the range which was set using the external alignment information. To place the external alignment at the top, we assigned it an alignment score which is 0.001 times larger than the best alignment score to all scores not initially placed at the top. This can be seen as a combination of **cSPA-amX** and **cSPA-A**.

- **nSPA-A**
  This **n**on-contiguous SPA aligner that simply returns the external alignment as the answer.

- **nSPA-AmX**
  This is our non-contiguous SPA aligner using external alignment information as

explained in Section 4.4.1. This returns multiple non-contiguous target candidates with the external alignment at the top.

- **nSPA-AmXr**
  This is a modified *nSPA-mA* that **r**emoves out-lier alignment links before the external alignments are used as described in Section 4.4.2.

All of these aligners return the multiple feature scores described in Section 4.4.3 so that they are combined in the decoder with the learned parameters(weights) in a separate parameter optimization stage. To get the external alignments, we ran the Moses toolkit on our training sets and used the *grow-diag-final* refinement method. We annotated the training sets with the alignment information so that SPA can use it in the EBMT system.

After that, we measured the translation performance differences between the contiguous SPA and non-contiguous SPA. The metrics we used to measure translation performance are BLEU by Papineni et al. (2002) and METEOR by Banerjee and Lavie (2005). We optimized parameters using Minimum Error Rate Training for BLEU on the development sets and measured performance on both BLEU and METEOR for the test sets. We set METEOR to do stemming and stemmed synonym matching.

For significance test, we used Paired Bootstrap Resampling by Koehn (2004b) with n=1000 [10].

4.5.1 Data

We investigated the phrasal aligners with three language pairs: Korean-English, Chinese-English , and French-English

Our Korean-English training data consists of 28,000 sentence pairs. Because these sentences are from conversations in the travel and business domains, they are much shorter than the Chinese-English and French-English sentences in sentence length. The development set *Dev* consists of 966 sentences with one reference, and the unseen test set *Unseen* has 2,170 sentences with one reference. Both test sets are in-domain.

[10] This is a paired t-test on n re-sampled data sets.

To build a Chinese-English training set, we drew sentence pairs with 70 or fewer words on the source side from the FBIS data. These are 341,636 sentence pairs, and the Chinese side was segmented using the Stanford Chinese segmenter by Chang et al. (2008). Our development set *Dev* consisting of 919 sentences was used as a test set in NIST Machine Translation Evaluation 2003, and our unseen test set *Unseen* is the newswire test set with 691 input sentences that was used in NIST Machine Translation Evaluation 2008. Both sets have 4 reference translations.

For French-English, we drew 300,000 sentence pairs from the Europal corpus. We used the dev2006 set as our development set *Dev* and the nc-test2007 as our unseen set *Unseen* from WMT 2006 and 2007 respectively. These test sets have one reference translation.

	sentences	source words	target words	references
Training set	28,034	248,263	266,583	1
Dev	966	8,591	-	1
Unseen	1,170	10,441	-	1

Table 4.2: Korean-English data

	sentences	source words	target words	references
Training set	341,636	9,155,903	11,571,657	1
Dev	919	42,946	-	4
Unseen	691	31,708	-	4

Table 4.3: Chinese-English data

Tables 4.2, 4.3, 4.4 describe the data we used in this evaluation for Korean-English, Chinese-English, and French-English, respectively.

	sentences	source words	target words	references
Training set	300,000	9,074,621	8,355,970	1
Dev	285	9,174	-	1
Unseen	2,007	58,168	-	1

Table 4.4: French-English data

4.5.2 Results and analysis

Korean-English results

Table 4.5 shows Korean-English results.

cSPA-mX shows that the number of alternatives that SPA returns are important. As the number of alternatives *X* increases, the BLEU score for *Dev* increases and it is the highest when *X* is 3. When *X* is 4, the BLEU score is slightly lower. This means that more alternatives are helpful in increasing translation performance, but too many alternatives may include noisy alternatives and make the system confused in discerning good alternatives for translation with given feature scores.

cSPA-amX shows about 0.5 BLUE score improvements over *cSPA-mX* for the same Xs. Because *cSPA-amX* determines a target range based on external word alignment, this shows that we had slight improvements by using external word alignment in determining target ranges although the improvements are not statistically significant with $0.05 < p < 0.1$. This shows that the word order difference in Korean and English sentences should be taken into account when we determine target ranges because *cSPA-mX* complete ignores word order differences.

cSPA-AmX is significantly better than *cSPA-mX* with $p < 0.0001$ and slightly better than *cSPA-amX* with $0.05 < p < 0.1$. The *cSPA-AmX* is also much better than *cSPA-A* looking at their BLEU scores. This means that the contiguous span of the external non-contiguous alignment is a good translation candidate and using it with the derived multiple alternative translations can increase the system performance.

nSPA-AmX is significantly better than *nSPA-A* as well with $p < 0.0001$. In this case, multiple non-contiguous alignments derived from the external non-contiguous alignment also increased the system performance. The *# SPA alts on Dev* denotes the number alternatives returned by non-contiguous SPA, which shows that although not many alternatives were added, they were very helpful because they increased the system performance significantly with $p < 0.0001$.

cSPA-AmX is slightly better than *nSPA-AmX* at their bests (i.e., *cSPA-Am5* and *nSPA-Am4*), but not significantly ($p = 0.146$). The system achieved comparable BLUE scores with *nSPA-AmX* although the number of alternatives is much smaller (i.e., the *# SPA alts on Dev* is much smaller). The differences of the best BLEU scores of *cSPA-AmX* and *nSPA-AmX* are not significant for both *Dev* and *Unseen*. Although *nSPA-AmX* did not outperform *cSPA-AmX*, it is still meaningful because its execution time was much shorter. We measured the execution times of *nSPA-Am4* and *cSPA-Am5* and they were 566 and 9,691 seconds.

Table 4.6 compares selected phrases for *nSPA-Am4* and *cSPA-Am7*. In this table, **SPC** denotes Selected Phrase Count, **ASPL** denotes Average Source Phrase Length, **ATPL** denotes Average Target Phrase Length, and **SNTPC** denotes Selected Non-contiguous Target Phrase Count. Overall the average phrase lengths are similar because only 5.9% of non-contiguous phrases are selected in decoding. The decoder did not interlock non-contiguous target phrases and this gave lower language model scores to the non-contiguous alignments. Note that the number of selected source words are less in the non-contiguous case. This is because the decoder did not use non-contiguous alignments that hurt hypothesis scores.

Analysis of score difference between BLEU and METEOR

In our translation experiments, we provided METEOR scores as well as BLEU scores. In general, METEOR scores are consistent with BLEU scores where we have significant improvements and this supports our improvements. For example, when we look at *cSPA-m3* and *cSPA-Am3* results, we see that for both Dev and Unseen sets, METEOR scores increased as BLEU scores increased. However, sometimes METEOR scores drop when BLEU scores increase. For example, when we look at *cSPA-Am4* and *cSPA-Am5*, the METEOR scores drop when the BLEU score increases. Table 4.7 shows how this happened. *cSPA-Am5* generated shorter hypotheses with higher precision and lower recall compared to *cSPA-Am4*. In METEOR, recall is weighted 9 times more than precision thus *cSPA-Am4* received a higher score than *cSPA-Am5*. But in BLEU, recall is not taken into account. Instead, the brevity penalty penalizes short hypotheses, but it did not affect the score enough to offset the higher precision of *cSPA-Am5*. In the table, the brevity penalty decreased the BLEU score only from 0.2615 to 0.2468 for *cSPA-Am5*. We had the same analytical results when we compared *cSPA-Am5* to *cSPA-Am6*. Another interesting point is the comparison of *cSPA-Am4* and *cSPA-Am6*. *cSPA-Am4* has higher precision and recall, but it has a lower BLEU score while it has a higher METEOR score. This is because it was penalized more by the shorter hypothesis length.

Chinese-English results

Table 4.8 shows Chinese-English translation results.

Firstly, as in the Korean-English translation results, *cSPA-m3* performed significantly better than *cSPA-m1* with $p < 0.0001$. We also observed a slight BLEU score drop when X is 4 as in the Korean-English results.

Secondly, *cSPA-amX* are better than *cSPA-mX*. The improvement of *cSPA-am4* over *cSPA-m3* is statistically significant with $p = 0.023$. This shows that there are significant word order mismatches between Chinese and English because *cSPA-mX* assumes the same word orders in the language pair.

Thirdly, *cSPA-am4* is better than *cSPA-A* and *cSPA-Am5* is better than *cSPA-am4* significantly with $p < 0.0001$. This shows that the external alignment itself is a very useful candidate as well as the derived multiple alternative candidates. The combined system *cSPA-AmX* outperforms any of the single system *cSPA-amX* and *cSPA-A*.

Fourthly, we used the external alignment itself as *nSPA*'s only translation candidate. This is denoted *nSPA-A* and performs significantly better than *cSPA-A*. However it is not as good as the best of *cSPA-AmX* because there is a better contiguous alignment candidate among the multiple contiguous candidates than the external non-contiguous alignment. We also derived multiple *nSPA* alternative translations (*nSPA-AmX*) but it did not outperform *cSPA-Am5* either. In fact, it underperformed *cSPA-Am5* because *nSPA* was not able to generate many translations. It created less than 1.3 candidates on average which did not give the translation system chances to outperform *cSPA-AmX*. Table 4.9 shows selected phrase statistics from our decoder. In this table, **SPC** denotes Selected Phrase Count, **ASPL** denotes Average Source Phrase Length, **ATPL** denotes Average Target Phrase Length, and **SNTPC** denotes Selected Non-contiguous Target Phrase Count. Non-contiguous phrases are only 1.6% and 7% for the Dev and Unseen sets respectively because the decoder cannot interlock non-contiguous target phrases which leads to lower language model scores.

Finally we removed outliers from the external alignment and investigated translation results (*nSPA-AmXr*). This approach did not help on *Dev*, but on *Unseen*, it slightly helped over *nSPA-AmX*. And the improvement was statistically significant with $p = 0.05$.

French-English results

Table 4.10 shows our French-English translation results. There are three observations we noticed for this language pair.

Firstly, the scores do not increase as the number of alternatives increases. Because French and English are very close languages in their structures, the external alignment may be accurate enough that the additional SPA translation alternatives derived from that are not helpful for translation.

Secondly, on *Dev*, *cSPA* and *nSPA* do not perform significantly differently in BLEU. Because the sentence structures of the two languages are similar, not many non-contiguous alignment are actually selected in decoding. Table 4.11 shows that only 3.5% and 3.2% of phrasal translations are non-contiguous for *Dev* and *Unseen* respectively. In this table, **SPC** denotes Selected Phrase Count, **ASPL** denotes Average Source Phrase Length, **ATPL** denotes Average Target Phrase Length, and **SNTPC** denotes Selected Non-contiguous Target Phrase Count.

Finally, *nSPA* performs significantly better on *Unseen*. Based on the average lengths of the selected source and target phrases in Table 4.11, we suspect the two data sets are different enough that *Unseen* takes more advantage with shorter phrasal translations by chance.

Summary

To summarize, firstly, returning multiple translation alternatives from SPA helps the system perform significantly better. The system performance increases as the number of alternatives increases up to 3 or 4 and then stays or decreases as more candidates come in to the search space. Secondly, using the external word alignment in determining target range is useful when language pairs are different in word order. In our experiments, we did have improvements with external word alignment for Korean-English and Chinese-English. But for French-English, which is a close language pair, we did not see improvements. Thirdly, in addition to multiple translation alternatives drawn from the target ranges determined based on external word alignment, we achieved more improvements when we used the external word alignment itself as the best candidate. Finally, non-contiguous alignment did not help the system performance. For Korean-English and Chinese-English, contiguous SPA performed better than non-contiguous SPA. However, for the French-English unseen set, non-contiguous SPA outperformed contiguous SPA.

Phrase Aligner	Dev		Unseen		# SPA alts on Dev
	BLEU	METEOR	BLEU	METEOR	
cSPA-m1	0.2231	0.4400	0.2331	0.4271	1
cSPA-m2	0.2306	0.4436	0.2410	0.4438	2
cSPA-m3	0.2346	0.4484	0.2414	0.4364	3
cSPA-m4	0.2336	0.4553	0.2422	0.4417	4
cSPA-am1	0.2284	0.4462	0.2425	0.4406	1
cSPA-am2	0.2350	0.4526	0.2484	0.4469	2
cSPA-am3	0.2393	0.4660	0.2532	0.4593	3
cSPA-am4	0.2415	0.4638	0.2507	0.4528	4
cSPA-am5	0.2396	0.4633	0.2522	0.4575	5
cSPA-A	0.2224	0.4421	0.2377	0.4410	1
cSPA-Am3	0.2412	0.4568	0.2536	0.4571	3
cSPA-Am4	0.2426	0.4722	0.2543	0.4674	4
cSPA-Am5	0.2468	0.4687	0.2552	0.4603	5
cSPA-Am6	0.2435	0.4720	0.2543	0.4638	6
nSPA-A	0.2289	0.4607	0.2452	0.4592	1
nSPA-Am3	0.2419	0.4654	0.2555	0.4656	1.240
nSPA-Am4	0.2430	0.4741	0.2539	0.4678	1.263
nSPA-Am5	0.2422	0.4702	0.2573	0.4693	1.270

Table 4.5: Korean-English results: BLEU/METEOR

	Dev		Unseen	
	nSPA-Am4	cSPA-Am5	nSPA-Am4	cSPA-Am5
SPC	4,447	4,502	5,316	5,320
ASPL	1.49	1.49	1.48	1.50
ATPL	1.61	1.64	1.64	1.66
SNTPC	263	0	312	0

Table 4.6: Korean-English selected phrase statistics in decoding

	cSPA-Am4	cSPA-Am5	cSPA-Am6
BLEU	0.2426	**0.2468**	0.2435
BLEU w/o Brevity Penalty	0.2435	0.2615	0.2435
METEOR	**0.4722**	0.4687	0.4720
Hyp. Length	**6,583**	6,153	6,615
Ref. Length	6,932	6,932	6,932
Precision	0.5501	**0.5761**	0.5459
Recall	**0.5224**	0.5114	0.5209

Table 4.7: BLEU and METEOR score comparison

Phrase Aligner	Dev		Unseen		# SPA alts on Dev
	BLEU	METEOR	BLEU	METEOR	
cSPA-m1	0.2000	0.4787	0.1664	0.4484	1
cSPA-m2	0.2247	0.5009	0.1831	0.4662	2
cSPA-m3	0.2307	0.5053	0.1864	0.4646	3
cSPA-m4	0.2279	0.5054	0.1860	0.4674	4
cSPA-am1	0.2075	0.4962	0.1711	0.4655	1
cSPA-am2	0.2247	0.5034	0.1866	0.4695	2
cSPA-am3	0.2314	0.4896	0.1866	0.4385	3
cSPA-am4	0.2355	0.4991	0.1922	0.4520	4
cSPA-am5	0.2351	0.5060	0.1912	0.4621	5
cSPA-A	0.2155	0.4988	0.1714	0.4698	1
cSPA-Am3	0.2346	0.5184	0.1980	0.4775	3
cSPA-Am4	0.2401	0.5177	0.1980	0.4831	4
cSPA-Am5	0.2423	0.5242	0.1996	0.4774	5
cSPA-Am6	0.2396	0.5057	0.1926	0.4594	6
nSPA-A	0.2352	0.5330	0.1785	0.4765	1
nSPA-Am3	0.2356	0.5371	0.1846	0.4932	1.243
nSPA-Am4	0.2377	0.5271	0.1864	0.4848	1.267
nSPA-Am5	0.2377	0.5377	0.1875	0.4945	1.274
nSPA-Am6	0.2356	0.5390	0.1878	0.4980	1.281
nSPA-Am3r	0.2364	0.5294	0.1860	0.4906	1.208
nSPA-Am4r	0.2373	0.5285	0.1903	0.4841	1.224
nSPA-Am5r	0.2358	0.5307	0.1908	0.4847	1.231

Table 4.8: Chinese-English results: BLEU/METEOR

	Dev		Unseen	
	nSPA-Am5	Am5	cSPA-nSPA-Am5	cSPA-Am5
SPC	14,216	14,480	10,170	10,323
ASPL	1.39	1.42	1.44	1.47
ATPL	1.55	1.58	1.57	1.62
SNTPC	240	0	164	0

Table 4.9: Chinese-English selected phrase statistics in decoding

Phrase Aligner	Dev		Unseen		# SPA alts on Dev
	BLEU	METEOR	BLEU	METEOR	
cSPA-m1	0.2378	0.5384	0.1912	0.5382	1
cSPA-m2	0.2385	0.5371	0.1902	0.5380	2
cSPA-m3	0.2354	0.5325	0.1895	0.5300	3
cSPA-m2	0.2335	0.5307	0.1901	0.5373	4
cSPA-am1	0.2359	0.5343	0.1907	0.5374	1
cSPA-am2	0.2320	0.5319	0.1878	0.5364	2
cSPA-am3	0.2360	0.5364	0.1895	0.5401	3
cSPA-am4	0.2365	0.5369	0.1932	0.5381	4
cSPA-A	0.2383	0.5377	0.1914	0.5423	1
cSPA-Am3	0.2407	0.5374	0.1930	0.5367	3
cSPA-Am4	0.2378	0.5377	0.1918	0.5316	4
cSPA-Am5	0.2409	0.5390	0.1924	0.5368	5
cSPA-Am6	0.2362	0.5375	0.1908	0.5346	6
nSPA-A	0.2450	0.5416	0.1874	0.5492	1
nSPA-Am3	0.2419	0.5420	0.2005	0.5444	1.244
nSPA-Am4	0.2416	0.5418	0.2003	0.5468	1.263
nSPA-Am5	0.2423	0.5425	0.2019	0.5489	1.268
nSPA-Am6	0.2412	0.5416	0.2026	0.5493	1.273

Table 4.10: French-English results: BLEU/METEOR

	Dev		Unseen	
	nSPA-Am5	cSPA-Am5	nSPA-Am5	cSPA-Am5
SPC	4,203	3,690	30,377	27,427
ASPL	2.16	2.49	1.85	2.09
ATPL	1.95	2.24	1.67	1.89
SNTPC	147	0	960	0

Table 4.11: French-English selected phrase statistics in decoding

Chapter 5

Chunk alignment

SPA finds translation phrases based on word translation probabilities. This means that the boundaries of the target phrases are determined by word translation probabilities. However, in the real world, we observe that a source phrase and a target phrase are a perfect translation pair even if they include words that do not have translational equivalents in the other side. For example, a Korean phrase ‘sa-moo-sil yi’ literally meaning ‘office NOM’ and an English phrase ‘the office’ are a good translation pair. But in word level, ‘yi’ does not have a translational equivalent in English and ’the’ does not have a translational equivalent in Korean. For this example case, although SPA may have the correct translation in the list of multiple alternative translations for the Korean phrase, SPA does not use linguistic knowledge to indicate it as a perfect translation.

However, if we consider each of the source and target phrases a unit and translate them as a unit, we can guarantee their correct translation. For this reason we investigate chunks as our basic translation units. The phrases above are legal chunks and show a nice example of translation by chunks. In this chapter, we discuss a new chunk alignment algorithm and methods for finding good chunk translation pairs.

5.1 High quality chunk

5.1.1 What is a chunk

Chunk is a linguistic concept pioneered by Abney (1991). A chunk is a non-recursive core of an intra-clausal constituent, extending from the beginning of the constituent to its head, but not including post-head dependents. A maximal chunk is a chunk that is contained in no other chunk and in this thesis work, we refer to a maximal chunk when we mention a chunk. Furthermore, chunks are defined strictly syntactically, not semantically, functionally, or lexically [1]. A typical chunk consists of a single content word surrounded by function words related to it. The order in which chunks occur is much more flexible than the order of words within chunks. When spoken, a strong stress will fall only once a chunk and pauses are most likely to fall between chunks.

1 Only a relative handful of such reports was received , the jury said , considering the widespread interest in the election , the number of voters and the size of this city .

2 Only [a relative handful] of [such reports] [was received] , [the jury] [said] , [considering] [the widespread interest] in [the election] , [the number] of [voters] and [the size] of [this city] .

3 [Only] [a relative handful] [of such reports] [was received] [,] [the jury] [said] [,] [considering] [the widespread interest] [in the election] , [the number] [of voters] [and] [the size] [of this city] [.]

The above example shows how Abney defined chunks and how we modified them for our machine translation purpose. The first sentence (“1”) shows the original sentence and the second sentence (“2”) shows chunking performed on the original sentence. Note

[1] Chunks are contiguous in most languages although there are non-contiguous chunks in some languages. In some cases, even non-contiguous chunks can be translated correctly. For example, in an English sentence ”The more I read the more tired my eyes get.”, ”the more” are really one disjoint chunk, but translations are usually fine when treated as two smaller separate chunks.

that adverbs and punctuation marks are not chunks. Neither are prepositions included in chunks. We define adverbs and punctuation marks as chunks and include prepositions into the following phrases. The third sentence (“3”) shows modified chunking by our modified chunk definition.

5.1.2 Advantages of using chunks

To begin, we discuss several advantages of using chunks as basic translation units. First, to some degree, we can systematically translate untranslatable tokens (words, morphemes) that exist only on one side of the language pair. These tokens can be translated properly using a phrase aligner such as SPA; however, additional efforts are needed to make the tokens selected in decoding because phrasal aligners may return multiple target candidate phrases and the correct translation may be one of them. Secondly, as chunks are n-gram phrases, they convey local reordering and context as well, although this advantage is also true for n-grams in phrasal translation. In addition, the number of chunks may better match across languages than the number of words, which may yield better alignment at the chunk level. Furthermore, because the order of chunks is more flexible than the order of words within a chunk, using chunks as blocks in translation has more flexibility in re-ordering than arbitrary n-grams crossing syntactic chunk boundaries. This is an important advantage when we translate from or into a language with relatively looser word-order than English or the Romance languages.

5.1.3 Uniqueness of our work

Our chunk-based work is different from previous work in the following ways:

First, our chunking is neither fully automatic nor bilingual. It exploits existing monolingual chunkers that use machine learning techniques to find chunk boundaries trained on a hand-annotated corpus with chunk boundaries. Most automated chunk detection algorithms heavily depend on human resources such as human dictionaries (Le et al., 2000; Hwang et al., 2004) and hand-written grammars (Wu, 1997) whereas others depend on co-occurrence statistics either bilingually or monolingually (Zhou et al., 2004; Watanabe et al., 2003). In this work, we use existing chunkers to avoid errors that can be caused by automatic chunking or insufficient bilingual resources. However, since we use monolingual chunkers, we do not maximize chunk correspondence between source and target languages.

Secondly, our work uses a new chunk alignment algorithm that is tightly combined with IBM word alignment models. In this chapter, we introduce a new chunk alignment algorithm. The basic idea is to apply the well-known IBM word alignment algorithms to chunk alignment by treating a chunk as a token and exploiting word translation probability to boost chunk alignment because a chunk is composed of multiple words. In other words, to alleviate data sparsity problems caused by using chunks as basic units, we will use word alignment information between a source chunk and a target chunk when we align them.

Third, in decoding, it combines target chunks as well as target fragments which are not chunks. Unlike the current EBMT system (Brown, 1996, 2005), this chunk-based system is a hybrid system that combines a typical string-based EBMT system and a chunk-based EBMT system. It uses a chunk as the basic translation unit when there is a good chunk level translation, otherwise it falls back to the string-based model.

5.2 Related work

Since translation by chunks can naturally add or remove words that exist only on one side of a language pair, some researchers have studied exploiting chunks in translation.

Le et al. (2000) used chunk alignment to get better word alignment. Given a dictionary and chunked English sentences, they made corresponding Chinese sentences chunked via chunk projection. More specifically, the citation for each word in an English chunk was found in the dictionary to discern its translation in the corresponding Chinese sentence. After resolving translation disambiguation using heuristics, the shortest Chinese word sequence including all the translation words is recognized as a chunk. The resulting Chinese chunk then becomes the translation of the English chunk.

Hwang et al. (2004) used chunk alignment to extract Korean dependency parse trees given Japanese dependency parse trees and a human dictionary. They first align words consulting a Japanese-Korean dictionary to find chunk boundaries and alignment and then they align the remaining words. They finally extracted bilingual knowledge from the aligned chunk pairs.

Zhou et al. (2004) extracted chunk pairs automatically to use in an SMT system. Their chunk detection is based on the assumption that the most co-occurrent word sequence may be a potential chunk. After aligning chunks using their co-occurrence similarity, they extract chunk pairs and report a significant improvement in translation quality.

Ma et al. (2007) studied an adaptable monolingual chunking approach. They learned word alignment in a parallel corpus and used this alignment information to find chunk boundaries in both languages.

Wu (1997) studied inversion transduction grammar (ITG) formalism for bilingual parsing for a parallel corpus. In this parse tree pair, the method naturally provides bilingual bracketing and alignment so that we can obtain aligned chunk pairs. However, it remains difficult to write a broad bilingual ITG grammar to deal with long sentences.

Watanabe et al. (2003) built a chunk-based statistical translation system. They deconstruct the translation model $P(J|E) = \sum_A P(J, A|E)$ to $P(J|E) = \sum_{\mathcal{J}} \sum_{\mathcal{E}} P(J, \mathcal{J}, \mathcal{E}|E)$ where $\mathcal{J}$ and $\mathcal{E}$ are the chunked sentences for J and E respectively. Then they deconstructed $P(J, \mathcal{J}, \mathcal{E}|E)$ further to $P(J, \mathcal{J}, \mathcal{E}|E) = \sum_A \sum_{\mathcal{A}} P(J, \mathcal{J}, A, \mathcal{A}, \mathcal{E}|E)$ where A is chunk alignment and $\mathcal{A}$ is word alignment for each chunk translation.

Koehn and Knight (2002) deconstructed a translation model into *sentence level reordering (SLR)*, *chunk mapping (CM)* and *word translations (W)*:

$$p(f|e) = p(SLR|e) \times \prod_i p(CM_i|e, SLR) \times \prod_j (W_{ij}|CM_i, SLR, e) \tag{5.1}$$

SLR defines how source and target chunks are connected and *CM* defines an alignment of source to target POSs. Finally *W* sets the lexical composition of the target language sentence. They reported improved performance over IBM Model 4 on a short sentence translation task.

Our approach uses monolingual chunkers and IBM word alignment models. For chunk alignment, we develop a new algorithm that uses word alignment information as chunk alignment evidence. To overcome lower chunk correspondence due to monolingual chunking, we use the *Refined Method* to find consistent chunk translation sequences that Och and Ney (2003) have used in phrasal translation detection. We explain this approach in more detail in Section 5.4.3.

5.3 Chunk detection

We first tried to detect chunks in a corpus automatically based on word co-occurrence statistics. However, due to the quality of our preliminary results and the difficulty of the task, we decided to use existing monolingual chunkers based on machine learning techniques that need some hand annotated training data for chunk boundaries.

In the next two subsections we describe the approach that we tried with the possibility for further investigation in the future. And in the final subsection of this section, we describe the monolingual chunkers we used.

5.3.1 Methods for deriving chunks and idiom information from corpora

Several methods have been proposed for deducing idiomatic phrases from corpora, both monolingually and bilingually.

Monolingually, Mutual Information (Cover and Thomas, 1991) is the commonly-used metric for determining the coherence of word sequences, while bilingually, cooccurrence counts are typically used.

Melamed (2001) uses (bilingual) Mutual Information (MI) to compute an objective function. However his method allows for only two words or compounds to be combined into a non-compositional compound, which may be insufficient to derive longer idiomatic phrases.

Often, the core of the meaning in a phrase (or sentence) is provided by one or more relatively infrequent – but highly salient – words. The same is commonly true for phrases or sentences that are translations of each other, even if, because of idiomatic usage, the kernel words themselves are not in translational correspondence. For example

Dutch	Het	regent	pijpenstelen
	[it]	[rains]	[pipe stems]
English	It	is raining	cats and dogs

One measure to approximate salience is the *Inverse Document Frequency* (IDF) by Jones (1972) which is commonly used in the Information Retrieval community:

$$\mathrm{IDF(w)} = \log(\frac{\mathrm{N}}{\mathrm{c_w}})$$

where w is the word or term under consideration, N is the corpus size (for our purpose, the number of sentences), and c_w is the number of sentences in which w occurs.

We also use IDF in detecting chunk boundaries and aligning the detected chunks. Our approach is described in the following:

1. Select words with $\mathrm{IDF(w)} > \theta$ as salient words, in both the source and the target languages.

2. Align salient words only.

3. Re-attach function words to salient words based on automatically derived linguistic rules.

First, we select salient words in a sentence using an IDF threshold. We consider these salient words the core meaning of each chunk [2]. Next, we align the salient words to align chunks. Finally we attach function words to salient words based on automatically derived linguistic rules to detect chunk boundaries. Our method to derive the rules is explained in Section 5.3.2.

[2] When there are consecutive salient words, we regarded them to be included in a single chunk.

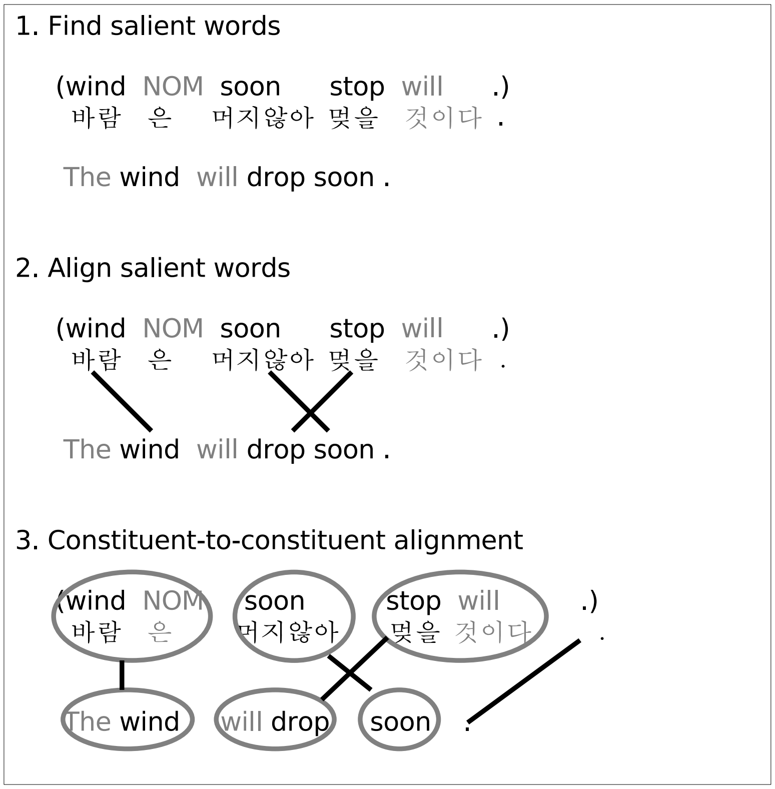

1. Find salient words
(wind NOM soon stop will .)
바람 은 머지않아 멎을 것이다 .
The wind will drop soon .
2. Align salient words
(wind NOM soon stop will .)
바람 은 머지않아 멎을 것이다 .
The wind will drop soon .
3. Constituent-to-constituent alignment
(wind NOM
바람 은
soon
머지않아
stop will
멎을 것이다
.)
.
The wind
will drop
soon
.

Figure 5.1: Constituent-to-constituent alignment

Figure 5.1 illustrates this method on an example. It first finds salient words. In this example, the black Korean words “wind”, “soon”, “stop” and the black English words “wind”, ”drop”, “soon” are salient words. Next, it aligns them. In this example, there are 3 salient words in each sentence and they are aligned. Finally, it attaches non-salient (function) words in gray to a salient word based on automatically derived chunk formation rules.

The threshold θ may be trained using any standard optimization algorithm such as hill climbing or simplex, using alignment accuracy (compared to a gold-standard human alignment) as an objective function.

5.3.2 Automatic derivation of chunk formation rules

Instead of assuming linguistic knowledge about each language in the pair (e.g., predominantly post-position, as in Korean or Japanese, or pre-position, as in English, of function words), these can be derived statistically from large monolingual corpora, which are readily available for most languages.

Instead of the normal calculation of collocation (Mutual Information with the following word, the following content word, or words in the other language), we now focus on MI between (classes of) high-IDF words and surrounding (classes of) low-IDF words. High-IDF words predominantly collocated with their right neighbor will by preference absorb post-positions, whereas those with higher MI with their left neighbor will favor pre-positions.

From a given corpus, rules can be derived that are either general over the corpus or specific to certain content words. In addition to Mutual Information, other metrics for phrasal cohesion can be explored. well.

5.3.3 Monolingual chunkers

We started experiments with the Chinese-English language pair. At the time we started the experiments, there was no Chinese chunker available to us. And because the data we had was already parsed, we decided to use the Chinese parse trees generated using Stanford parser (Klein and Manning, 2003b,a). We wrote a simple program that splits a parse tree into chunks. We took the same strategy for the English side because English sentences were also already parsed.

Next, we did experiments with the Korean-English pair. Like Chinese, we could not find a Korean chunker and wrote a rule-based Korean chunker that makes use of Part-Of-Speech tags returned by the Korean morphological analyzer we used (Shim and Yang). For English, instead of asking the Stanford parser to parse sentences and recognize chunks, we used an existing monolingual chunker. We used SNoW shallow parser (Carlson et al., 1999) for English.

When we later began experiments on the French-English language pair, we found and used TreeTagger (Schmid, 1994, 1995) for French. Because it also supported English chunking, we decided to use it for English chunking n the language pair as well. A simple program was written to extract chunkers from its hierarchical structures of results.

Note that all three language pairs include English but each used different chunkers. We hoped that monolingual chunkers developed by the same developer were designed with the concept of chunk, although we do not prove this assumption in this work.

5.4 Chunk alignment

In general, aligning chunks is a harder task than aligning words on the same training data set if we use an unsupervised method such as IBM Model 4. The reason is that by using chunks as a basic unit, we have much less evidential statistics than we do when we use words as basic units.

For example, “in the office” is a chunk and appears much less than each of the comprising words “in”, “the” and “office” in a corpus. The statistical evidence for aligning the chunk is less obvious than that of aligning the comprising words and this results in poorer alignment quality for chunks. Hence aligning words and using this alignment information in chunk alignment is an important idea unless we have a gigantic corpus in which statistical evidence for chunk alignment is reasonably sufficient. But in reality, it is hard to build such an enormous corpus. Instead, we investigate a new method that induces chunk alignment from word alignment together with chunk co-occurrence statistics.

5.4.1 The baseline system

Our baseline system is simply the Moses alignment system that regards chunks as basic translation units and performs bi-directional alignment. For that, we concatenate all the member words in a chunk and run Moses on word-concatenated chunks. The problem with this method is that when the data becomes sparser, the system has less statistical evidence for chunk alignment. For this reason, chunk alignment quality may become poorer than word alignment quality.

5.4.2 Word-mapping-based chunk alignment

To overcome the data sparseness problem in the baseline system, we first perform word alignment and align a chunk pair when there is at least one word mapping among the source and target words in the chunk pair. Formally,

- Let $\mathbf{f}$ and $\mathbf{e}$ be chunks and $\mathbf{f}$ be $f_1^n = f_1 f_2 ... f_n$ and $\mathbf{e}$ be $e_1^m = e_1 e_2 ... e_m$.
- $\mathbf{f}$ and $\mathbf{e}$ are aligned if there exist any word alignment (f_i,e_j) where $1 \leq i \leq n$ and $1 \leq j \leq m$.

In the first stage, we align words by running GIZA++ on a training corpus bi-directionally in Moses. And then we find chunk boundaries monolingually. Finally, we align chunks based on word mappings for the words in a chunk pair. This method compensates for the data sparseness problem to some degree.

However, this approach only counts word mappings and ignores chunk level statistics and fertility. Fossum et al. (2008) also pointed out that function words that do not have translational equivalents can be aligned to function words, which in this case, can produce erroneous chunk alignments.

5.4.3 Using GIZA++ for chunk alignment

To take advantages of the baseline system and word-mapping-based chunk alignment system, we designed a hybrid system. In the hybrid system, we first concatenate all the words in chunks in a specially designed way (i.e., we can place a special delimiter character in between words belonging to the same chunk) to use as basic units in GIZA++. Next, we modify GIZA++ to take the source and target chunks and a statistical word translation dictionary as input. The modified GIZA++ uses the dictionary to re-weight chunk translation evidence by word translations within chunk pairs.

- Let $\mathbf{f}$ and $\mathbf{e}$ be chunks and $\mathbf{f}$ be $f_1^n = f_1 f_2 ... f_n$ and $\mathbf{e}$ be $e_1^m = e_1 e_2 ... e_m$.
- $T(\mathbf{f}|\mathbf{e})$ in IBM models is

$$T(\mathbf{f}|\mathbf{e}) = \frac{C(\mathbf{f}, \mathbf{e})}{\sum_k C(\mathbf{f}_k, \mathbf{e})} \tag{5.2}$$

- We redefine it as,

$$T(\mathbf{f}|\mathbf{e}) = \frac{C'(\mathbf{f}, \mathbf{e})}{\sum_k C'(\mathbf{f}_k, \mathbf{e})} \tag{5.3}$$

  where

$$C'(\mathbf{f}, \mathbf{e}) = C(\mathbf{f}, \mathbf{e}) \times F(\mathbf{f}, \mathbf{e}) \tag{5.4}$$

  where $F(\mathbf{f}, \mathbf{e})$ is a weighting function and for this, we used *power means* with power $p = 2$ [3]:

$$F(\mathbf{f}, \mathbf{e}) = \left(\frac{1}{m} \sum_{j=1}^{m} max_{i=1}^{n} (T(f_i|e_j))^p \right)^{\frac{1}{p}} \tag{5.5}$$

  .

5.4.4 Word alignment boosted by character n-gram

Like word boosting in chunk alignment, we can also use character n-grams in a word to boost word alignment. This technique is most helpful when dealing with morphologically-rich languages for which parallel data is insufficient. Frequently, to make parallel data correspond at the word level, we apply a morphological analyzer. Still, this technique has its own problems. We may be unable to find a corresponding token in the other side for a morpheme. It is also difficult to decide which level of analysis is adequate. For example, an inflected Korean verb, often, has more than 5 morphemes, but the corresponding English tokens number only two or three. In this case, using character n-grams as pseudo

[3] The power mean is also known as a generalized mean with exponent p. Depending on the p value, it can be a minimum ($p = -infinity$), harmonic mean($p = -1$), geometric mean($p = 0$), arithmetic mean($p = 1$), quadratic mean($p = 2$) or maximum($p = infinity$) and this variation allows us to efficiently investigate which mean works the best. We empirically chose $p = 2$ to maximize our chunk alignment accuracy.

morphemes rather than completely splitting Korean morphemes can produce an improvement in word alignment. Of course, the best case will be when we can use real morphemes instead of character n-grams.

In this case, the formula will be the same as Equation 5.4, but instead of using word translation probability, we use character n-gram translation probability, which is trained separately.

We employ this approach in our Korean-English translation experiments. To obtain character n-gram translation probability, we replace Korean tokens that are 4 words or longer with character bi-grams from them and English tokens that are also 4 words or longer with character 4-grams from them. These were empirically set up by looking at alignment results.

5.4.5 Iterative refinement

Kim and Vogel (2007) showed that word alignment and phrasal alignment can help each other. By giving back phrasal alignment information to the word aligner, they built a better lexicon, and this improvement on word alignment produced a better phrasal alignment in turn. This is applicable to our chunk alignment since chunks are also phrases (n-grams). This is particularly beneficial for word alignment improvement because we have strict chunk boundaries that prevent a word aligner from mapping words crossing chunk boundaries.

A simple way to investigate this technique is to iterate the two steps until convergence is reached. Formally we start with iteration $i = 0$, performance $Q_0 = 0$, corpus $C_0 = C$ (the initial corpus), and aligned chunk sequence pairs $P_0 = \phi$.

1. $i \leftarrow i + 1$
2. We add aligned chunk sequence pairs to the corpus to update it: $C_i = C_0 \cup P_{i-1}$.
3. We align words in C_i and calculate alignment quality Q_i.
4. We stop if $Q_i - Q_{i-1} < \epsilon$

5. We align chunks in the original corpus C_0 and extract aligned chunk sequence pairs P_i.

6. We return to step 1.

In step 5, we used the Moses alignment system that works with the modified GIZA++ for our word-boosted chunk alignment model.

5.5 Evaluation

5.5.1 Metric

In this evaluation, we measure precision, recall and F_1 for chunk alignment. When we have hand-aligned target chunks $H = \{\mathbf{h}_j | j = 0, ..., l\}$ and target chunks found $M = \{\mathbf{m}_k | k = 0, ..., m\}$ by a chunk alignment algorithm for each source chunk $\mathbf{f}_i$, we calculate precision $P = \frac{|H \cap M|}{|M|}$ and recall $R = \frac{|H \cap M|}{|H|}$. Note that, unlike SPA alignment accuracy evaluation, we did not exclude target chunks that are in M but not aligned in the hand-aligned corpus We decided to include them this time because they are actually passed to our decoder and used in translation.

5.5.2 Systems compared

We essentially compared three chunk alignment systems. All the systems are trained using the Moses alignment system.

1 *Baseline*: This is the pure chunk-based system. We concatenated all the words in a chunk in the training and test sets.

2 *Word-map*: This system is the word-mapping-based chunk alignment system. Because *Baseline* is very weak due to data sparseness, we consider this system to be our actual baseline system to which we will compare our new approach.

3 *Word-boost*: This is our new approach in which chunk translation probabilities are weighted by word translations by using our modified *GIZA++*.

For the *Baseline* system, we used the *grow-diag-final (G-D-F)* heuristic when we combined both directional alignments. For the *Word-map* system, we compared different refinement heuristics such as the *grow (Grow)*, *grow-diag (G-D)*, *grow-diag-final (G-D-F)*, *union (Union)*, and *intersect (Intersect)*. For the *Word-boost* system, we compared different refinement heuristic combinations for both word alignment and chunk alignment.

5.5.3 Data

We use French-English and Chinese-English language pairs to compare the alignment algorithms in alignment accuracy.

For French-English, we use 300,000 sentence pairs as the training set and 37 sentence pairs as the hand-aligned set. The training set was drawn from Canadian Hansards and the hand-aligned corpus was obtained from ACL WMT 2008.

In the training set, the French sentences are an average of 14.3 chunks and 25.7 words long and the English sentences are an average of 14.3 chunks and 24.5 words long. The chunks are 1.8 words long and 1.7 words long on average in French and English respectively. Table 5.1 describes the training data we use in this experiment.

	sentences	chunks	words
French	300,000	4,282,828	7,706,060
English	300,000	4,292,017	7,347,401

Table 5.1: Training set for French-English

Table 5.2 shows the French-English hand-aligned set we use. The data is originally hand-aligned at the word level but we derive a chunk-aligned set by aligning chunk pairs whenever any word pair in a chunk pair is aligned by human exactly as in *Word-map*. In French, the sentences have an average of 10.6 chunks and 19.5 words, and the chunks are an average of 1.8 words long. The English sentences have an average of 10.9 chunks and 17.9 words and the English chunks are an average of 1.6 words long. One thing to note in this table is that for French-English, word alignment has relatively high fertility (i.e., the number *4+ link* is very large) because the human aligner was not able to come up with word to word alignment in many cases and just aligned a multi-word phrase to a multi-word phrase which are then fully aligned at the word level.

	sentences	unit	count	0 link	1 link	2 link	3 link	4+ link
French	37	word	721	43	366	93	34	185
		chunk	392	27	218	49	42	56
English	37	word	661	35	296	91	52	187
		chunk	403	18	217	79	39	50

Table 5.2: Hand-aligned corpus for French-English

For Chinese-English evaluation, we use the same training set as in Chapter 4 and the same hand-aligned set as in Chapter 3.

On average, in the training set, the Chinese sentences consist of 26.8 words and 18.1 chunks in which the chunks are composed of 1.5 words. Likewise, the English sentences consist of an average of 33.9 words and 18 chunks in which the chunks are 1.8 words long. Table 5.3 describes the Chinese to English training data. Both the French-English and Chinese-English language pairs show that the chunk level correspondence is higher than the word level correspondence.

	sentences	chunks	words
Chinese	341,636	6,177,252	9,155,903
English	341,636	6,419,184	11,571,835

Table 5.3: Training set for Chinese-English

Table 5.4 shows the Chinese-English hand-aligned set. The chunk-aligned version is obtained in the same way as the French-English chunk aligned version is obtained. In

Chinese, the sentences have an average of 9.6 chunks and 13.8 words, and the chunks are an average of 1.4 words long. On average, the English sentences have 9.3 chunks and 16.1 words, and the English chunks are 1.7 words long. Unlike the French-English hand-aligned corpus, this corpus does not have as many high fertility words.

	sentences	unit	count	0 link	1 link	2 link	3 link	4+ link
Chinese	386	word	5,337	1,419	3,316	544	55	3
		chunk	3,721	1,112	2,095	372	96	46
English	386	word	6,277	2,329	3,393	488	60	7
		chunk	3,592	1,155	1,797	459	120	61

Table 5.4: Hand-aligned corpus for Chinese-English

5.5.4 Results and analysis

Chunk alignment

Table 5.5 shows French-English chunk alignment performance. Our secondary baseline *Word-map* performs much better than the original *Baseline*. This difference in performance can be explained by *Word-map*'s basis on word alignment, for which we have much better statistical evidence for alignment, compared to *Baseline*'s sole reliance on chunk co-occurrences. Our approach shows better overall scores than the strong baseline *Word-map* and is the best with *G-D* for word alignment and *G-D-F* for chunk alignment.

Table 5.6 shows Chinese-English chunk alignment performance. The Chinese-English results show a similar trend to the French-English alignment accuracy results. *Word-map* is a much stronger baseline and *Word-boost* outperforms it in the best case with *G-D* & *G-D-F* or *G-D* & *Union* for word alignment and chunk alignment respectively.

For the following experiments on chunk-based translation, we used *G-D* & *G-D-F* for word alignment and chunk alignment respectively because this combination gives the best performances for both language pairs.

System	Word H.	Chunk H.	Recall	Prec.	F_1	$\|M\|/\|H\|$
Baseline	N/A	G-D-F	0.4753	0.5599	**0.5141**	0.8489
Word-map	Grow	Link	0.4629	0.7125	0.5612	0.6497
Word-map	G-D	Link	0.5247	0.7461	0.6161	0.7033
Word-map	G-D-F	Link	0.6085	0.7111	**0.6558**	0.8558
Word-map	Union	Link	0.6250	0.6791	0.6509	0.9203
Word-boost	G-D	G-D	0.5659	0.8158	0.6683	0.6937
Word-boost	G-D	G-D-F	0.5865	0.8196	**0.6837**	0.7157
Word-boost	G-D	Union	0.5934	0.7985	0.6809	0.7431
Word-boost	G-D-F	G-D-F	0.5852	0.8068	0.6783	0.7253
Word-boost	Union	G-D-F	0.5852	0.8161	0.6816	0.7170
Word-boost	Union	Union	0.6003	0.7874	0.6812	0.7624

Table 5.5: Chunk alignment results for French-English

Table 5.7 shows significance test results for French-English and Chinese-English. The systems compared are *Word-map*, the word-link based chunk alignment which uses G-D-F for word alignment (S1) and *Word-boost*, the chunk alignment system which uses G-D for word alignment and G-D-F for chunk alignment (S2). We do not compare the baseline system because it is obviously significantly worse for both language pairs.

For both language pairs, we compare F_1 of the alignment results by the two systems for each source chunk. After removing tied chunks, we calculate paired t-test and obtained p of 0.001 and 0.0001 for French-English and Chinese-English, respectively.

Iterative refinement

Table 5.8 shows word and chunk alignment performance with an iterative refinement approach on French-English. In the second iteration, the results show that *Word-map* and word alignment (*Word-align*) improved significantly , which is what we expect when applying iterative refinement. However, we do not see improvement on chunk alignment.

System	Word H.	Chunk H.	Recall	Prec.	F_1	$\|M\|/\|H\|$
Baseline	N/A	G-D-F	0.4519	0.3105	**0.3681**	1.4552
Word-map	G-D	Link	0.4993	0.4228	0.4579	1.1807
Word-map	G-D-F	Link	0.5712	0.4080	**0.4760**	1.4001
Word-boost	Intersect	Intersect	0.4022	0.4125	0.4073	0.9748
Word-boost	Grow	Grow	0.4456	0.4052	0.4244	1.0998
Word-boost	G-D	G-D	0.5367	0.4577	0.4941	1.1726
Word-boost	G-D	G-D-F	0.5538	0.4582	**0.5015**	1.2086
Word-boost	G-D	Union	0.5625	0.4525	**0.5015**	1.2430
Word-boost	G-D-F	G-D-F	0.5514	0.4530	0.4974	1.2173
Word-boost	Union	Union	0.5835	0.3911	0.4683	1.4918

Table 5.6: Chunk alignment results for Chinese-English

Lang. pair	chunks	S1 $>$ S2	S1 $=$ S2	S1 $<$ S2	p
French-English	392	37	292	63	0.001
Chinese-English	3,718	164	3,196	358	0.0001

Table 5.7: Significance tests

The score actually decreased slightly, but not significantly. This decrease probably occurs because the constraints that the chunk aligner gives to the word aligner are stronger than the constraints that the word aligner gives to the chunk aligner. In other words, the chunk aligner encourages the word aligner to respect chunk boundaries, but the word aligner encourages the chunk aligner to respect word alignment inside already aligned chunk pairs in the previous iteration which may not have a great impact.

The fact that we improve word alignment is very noteworthy because we are building a translation system that uses both chunk alignment and word alignment. Word alignment will be utilized in translation when sufficiently good chunk alignments are absent.

Table 5.9 shows iterative word and chunk alignment performance with modified recall

Iteration	System	Word H.	Chunk H.	Recall	Prec.	F_1	$\lvert M\rvert/\lvert H\rvert$
Chunk alignment							
1	Word-map	G-D	Link	0.5247	0.7461	0.6161	0.7033
	Word-map	G-D-F	Link	0.6085	0.7111	**0.6558**	0.8558
	Word-boost	G-D	G-D-F	0.5865	0.8196	**0.6837**	0.7157
2	Word-map	G-D	Link	0.5508	0.7786	0.6452	0.7074
	Word-map	G-D-F	Link	0.6071	0.7530	**0.6722**	0.8063
	Word-boost	G-D	G-D-F	0.5865	0.8057	**0.6789**	0.7280
	Word-boost	G-D-F	G-D-F	0.5838	0.8080	0.6778	0.7225
Word alignment							
1	Word-align	Grow	N/A	0.2479	0.5676	0.3451	0.4368
	Word-align	G-D	N/A	0.2947	0.6547	0.4065	0.4501
	Word-align	G-D-F	N/A	0.3415	0.6934	**0.4576**	0.4925
	Word-align	Union	N/A	0.3543	0.6632	0.4541	0.5343
2	Word-align	G-D	N/A	0.3304	0.7127	0.4515	0.4635
	Word-align	G-D-F	N/A	0.3655	0.7412	**0.4896**	0.4930

Table 5.8: Iterative refinement results for French-English

on French-English alignment. In Table 5.8, we saw that word alignment F_1 scores are significantly lower than chunk alignment F_1 scores. The lower scores are the result of word alignment's much lower recall due to high word fertility in the hand-aligned set, as shown in Table 5.2. Chunk alignment recall alleviates this problem because usually consecutive words are split into fewer chunks. Therefore, to make them more comparable, we modify our recall calculation for each source unit: $R = 1$ if $|H \cap M| > 0$ and 0 otherwise.

Table 5.10 shows word and chunk alignment performance using an iterative refinement approach on Chinese-English alignment. As seen in French-English alignment evaluation, the results from the second iteration show that *Word-map* and word alignment (*Word-align*) improved significantly, but chunk alignment does not.

Iteration	System	Word H.	Chunk H.	Recall	Prec.	F_1	$\|M\|/\|H\|$
Chunk alignment							
1	Word-map	G-D	Link	0.8411	0.7461	0.7908	1.4027
	Word-map	G-D-F	Link	0.9342	0.7111	**0.8075**	1.7068
	Word-boost	G-D	G-D-F	0.9479	0.8196	**0.8791**	1.4274
2	Word-map	G-D	Link	0.8849	0.7786	0.8284	1.4110
	Word-map	G-D-F	Link	0.9370	0.7530	**0.8350**	1.6082
	Word-boost	G-D	G-D-F	0.9370	0.8057	0.8664	1.4521
	Word-boost	G-D-F	G-D-F	0.9370	0.8080	**0.8677**	1.4411
Word alignment							
1	Word-align	Grow	N/A	0.6077	0.5676	0.5870	1.1563
	Word-align	G-D	N/A	0.7168	0.6547	0.6844	1.1917
	Word-align	G-D-F	N/A	0.8230	0.6934	**0.7527**	1.3038
	Word-align	Union	N/A	0.8333	0.6632	0.7386	1.4145
2	Word-align	G-D	N/A	0.7847	0.7127	0.7470	1.2271
	Word-align	G-D-F	N/A	0.8599	0.7412	**0.7962**	1.3053

Table 5.9: Iterative refinement results with relative recall for French-English

Comparison to SPA

As mentioned in the introduction, SPA can detect chunk translation as well. Given a source chunk, SPA returns a list of possible translations and the correct target chunk can be included in the list.

To see how *Word-boost* compares to SPA, we compared their alignment performance for source chunks. We first categorized the source chunks into three classes.

- **Chunk**: all the single source chunks.

- **C_Chunk_H**: the source chunks that are aligned consistently by humans. An explanation of consistency can be found in Och and Ney (2003) or Section 6.2.1.

Iteration	System	Word H.	Chunk H.	Recall	Prec.	F_1	$\|M\|/\|H\|$
Chunk alignment							
1	Word-map	G-D	Link	0.4993	0.4228	0.4579	1.1807
	Word-map	G-D-F	Link	0.5712	0.4080	**0.4760**	1.4001
	Word-boost	G-D	G-D-F	0.5538	0.4582	**0.5015**	1.2086
2	Word-map	Grow	Link	0.4810	0.4252	0.4514	1.1313
	Word-map	G-D	Link	0.5199	0.4357	0.4741	1.1933
	Word-map	G-D-F	Link	0.5715	0.4321	**0.4921**	1.3224
	Word-boost	G-D	G-D-F	0.5514	0.4591	**0.5010**	1.2011
	Word-boost	G-D-F	G-D-F	0.5535	0.4576	**0.5010**	1.2095
Word alignment							
1	Word-align	Grow	N/A	0.5816	0.7655	0.6610	0.7598
	Word-align	G-D	N/A	0.6417	0.7287	**0.6824**	0.8806
	Word-align	G-D-F	N/A	0.7105	0.6546	0.6814	1.0855
	Word-align	Union	N/A	0.7289	0.6191	0.6696	1.1773
	Word-align	Intersect	N/A	0.4852	0.8877	0.6275	0.5466
2	Word-align	Grow	N/A	0.6263	0.7355	0.6765	0.8516
	Word-align	G-D	N/A	0.6767	0.7127	0.6942	0.9496
	Word-align	G-D-F	N/A	0.7320	0.6704	**0.6998**	1.0918

Table 5.10: Iterative refinement results for Chinese-English

- **C_Chunk_W**: the source chunks that are aligned consistently by *Word-boost*.

And for each of them, we compared word level *F1* of three chunk/phrase alignment algorithms: *SPA*, *SPA(Top-10)* and *Word-boost*. *SPA(Top-10)* picks the oracle alignment (alignment with the best *F1*) in the top-10 list from SPA.

Table 5.11 and Table 5.12 show word level *F1* of the three aligners for Chinese-English, and French-English and Figures 5.2 and Figure 5.3 show them in bar graphs.

For *Chunk*, *SPA* and *SPA(Top-10)* performs much better than *Word-boost* for both language pairs. This means that *Word-boost* tends to have a lot of non-one-to-one chunk align-

Chunk type	Count	Aligner	Recall	Precision	F1	Len(M)/Len(H)
Chunk	3,718	SPA	0.6332	0.6913	0.6610	0.9160
		SPA(Top-10)	0.9174	0.8170	0.8643	1.1230
		Word-boost	0.8271	0.4622	0.5930	1.7893
C_Chunk_H	1,752	SPA	0.7243	0.7890	0.7552	0.9108
		SPA(Top-10)	0.9584	0.8968	0.9266	0.1069
		Word-boost	0.9080	0.6950	0.7874	1.3064
C_Chunk_W	2,313	SPA	0.7141	0.7631	0.7378	0.9357
		SPA(Top-10)	0.9525	0.8532	0.9001	1.1163
		Word-Boost	0.8634	0.7286	0.7903	1.1850

Table 5.11: Chinese-English: Word alignment accuracy by SPA, SPA(Top-10) and *Word-boost*

ments and returns all the linked target chunks for them. This causes lower precision values and consequently leads to lower *F1* values. For the same reason, the *Len(M)/Len(H)* value for *Word-boost* is much higher than those of the others.

For consistently aligned source chunks *C_Chunk_H* and *C_Chunk_W*, *Word-boost* outperforms *SPA*. But it performs worse than *SPA(Top-10)* which is an oracle alignment for the top-10. It performs close to *SPA(Top-10)* for French-English while the differences are larger for Chinese-English.

The results show that for consistently aligned source chunks, *Word-boost* performs much better than *SPA* although it performs worse than *SPA(Top-10)*. This leads us to use *Word-boost* for consistently aligned source chunks by *Word-boost* in translation. Note that *C_Chunk_W* is obtainable while *C_Chunk_H* is not. *C_Chunk_H* is available only from a hand-aligned corpus. On the other hand, the *Count* columns in both tables show that the coverage with the consistently aligned source chunks drops substantially. This means that *SPA* should play a very important role in finding translations for the uncovered chunks.

Note that *SPA(Top-10)* is the oracle alignment in the top-10 list and its *F1* is higher than that of *Word-boost*. This implies that *SPA(Top-10)* has the potential to outperform

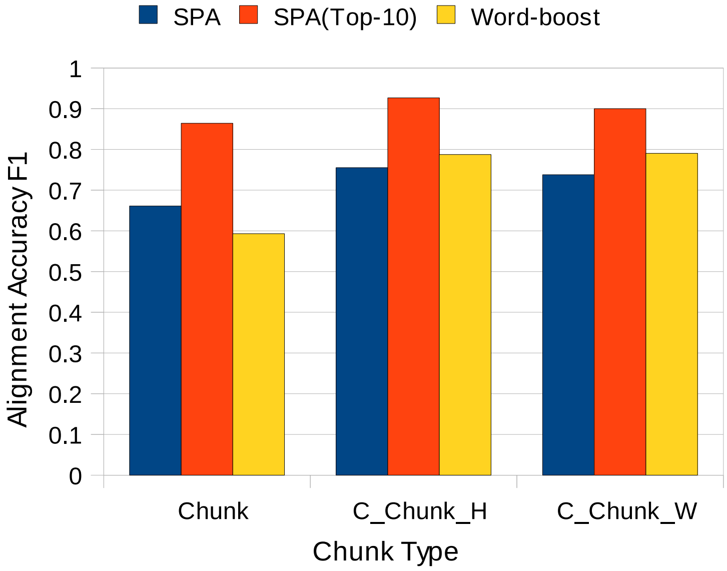


Figure 5.2: Chinese-English: Word alignment accuracy by SPA, SPA(Top-10) and *Word-boost*

Word-boost in translation.

Summary

To summarize, our new method *Word-boost* improves chunk alignment quality significantly over our strong baseline *Word-map*. These improvements are consistent through the different language pairs, Chinese-English and French-English. Furthermore, when we use chunk alignment to help word alignment, we find significant improvements on word alignment. These improvements are also consistent for both Chinese-English and French-English alignment.

We also compared *Word-boost* to *SPA* and showed that it performs better for the consistently aligned source chunks in alignment accuracy. This guides us to use *Word-boost* alignment for consistently aligned source chunk matches in translation.

In the next chapter, based on our chunk alignment analysis, we investigate a hybrid

Chunk type	Count	Aligner	Recall	Precision	F1	Len(M)/Len(H)
Chunk	392	SPA	0.4441	0.6951	0.5420	0.6390
		SPA(Top-10)	0.6828	0.8206	0.7454	0.8320
		Word-boost	0.4044	0.4114	0.4078	0.9831
C_Chunk_H	216	SPA	0.8202	0.7820	0.8007	1.0489
		SPA(Top-10)	0.9180	0.8946	0.9061	1.0262
		Word-boost	0.9511	0.8321	0.8876	1.1431
C_Chunk_W	242	SPA	0.6107	0.8113	0.6968	0.7527
		SPA(Top-10)	0.7746	0.9014	0.8332	0.8593
		Word-boost	0.6708	0.9009	0.7690	0.7445

Table 5.12: French-English: Word alignment accuracy by SPA, SPA(Top-10) and *Word-boost*

translation system that uses *Word-boost* for chunk-based translation and SPA for phrasal translation.

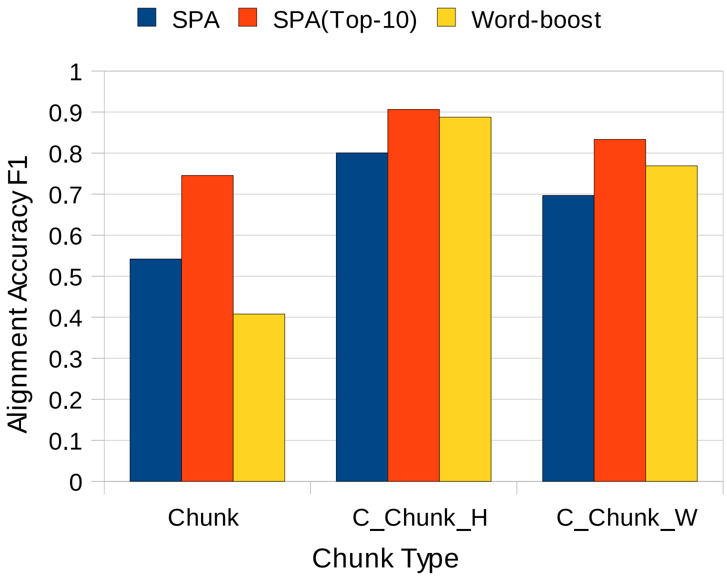


Figure 5.3: French-English: Word alignment accuracy by SPA, SPA(Top-10) and *Word-boost*

Chapter 6

Chunk-Based translation

In Chapter 5, we detected chunks and aligned them in a parallel corpus. In this chapter, we describe how we use the aligned chunks in translation.

We first explain how we extract consistent chunk sequence translations and assign them feature scores to be used in translation.

We next investigate chunk fuzzy matching. as an effort to overcome the unknown chunk problem As mentioned before, we usually have a higher unknown unit rate for chunks than for words because a chunk is a combination of one or more words. Although the chunk fuzzy matching helps the unknown chunk problem, we still have a significant number of unknown chunks because the chunk fuzzy matching is not sufficient to cover a significant number of unknown chunks.

For this reason, we will eventually need the word/phrase-based CMU EBMT system that provides translations for chunks that have very poor translations or for which we cannot find translations. We finally describe a system that combines SPA with a chunk-based translation system.

6.1 Baseline system

By using the SPA phrasal aligner in the CMU EBMT system, we achieved significant improvements in translation performance. Since then, we have been investigating further approaches for continued improvement of the EBMT system with SPA. Naturally, we use our CMU EBMT system with SPA as our baseline system in this work as well.

6.2 Chunk-based system

6.2.1 Consistent chunk alignment

In the previous chapter, we investigated algorithms to align chunks in parallel texts. However, the source and target sentence chunks we use are not detected in a synchronous way. Therefore, as in word alignment, we have a lot of one-to-many, many-to-one, and many-to-many relationships between source chunks and target chunks. For this reason, we need to find consistently aligned chunk sequence pairs [1] as translation pairs using the *Refined Method* that Och and Ney (2003) used for phrase extraction. We explain this using the version implemented by Koehn in Moses. We start with the intersection of the two chunk alignments adding new alignment points that exist in the union of two chunk alignments and connecting at least one previously unaligned chunk. First, we expand only to alignment points that are directly adjacent. We check for potential alignment points starting from the top right corner of the alignment matrix, checking for alignment points for the first target chunk, then continuing with alignment points for the second target chunk, and so on. We iterate this until we find no more alignment points to add. In the final step, we add non-adjacent alignment points the same requirements with the exception of adjacency.

We collect all aligned chunk sequence pairs that are consistent with the chunk alignment: Only the chunks in a legal chunk sequence pair are aligned to each other, and not to chunks outside. In our translation, if there is a partial match from the source side of an

[1] The chunks in a legal chunk sequence pair are only aligned to each other and not to chunks outside.

atomic legal chunk sequence pairs [2] , we do not use this chunk alignment because chunk alignment is not consistent in such a case.

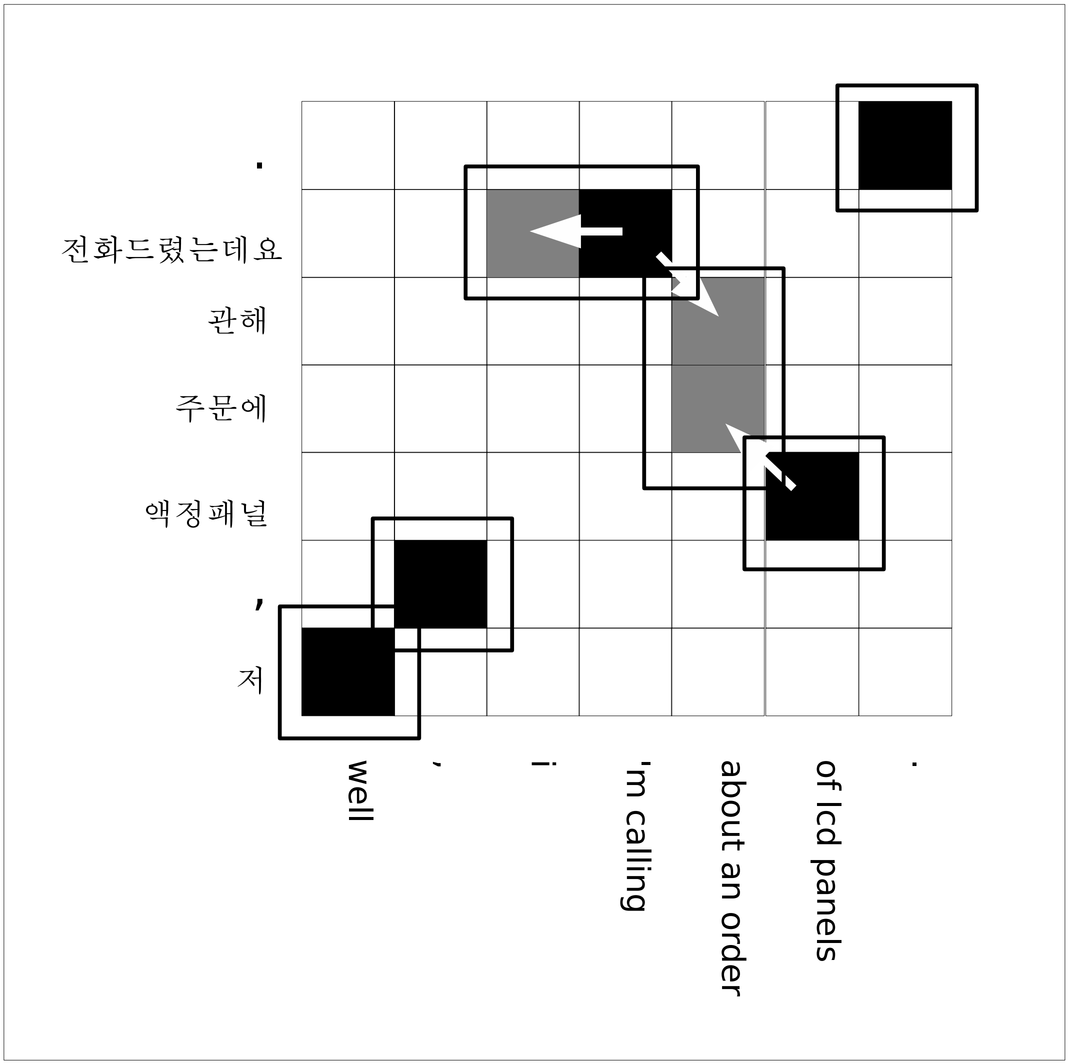


Figure 6.1: Chunk translation sequence pair extraction

Figure 6.1 illustrates how the *Refined Method* refines chunk alignment for machine-detected chunks and how chunk translation sequence pairs are extracted afterwards on a

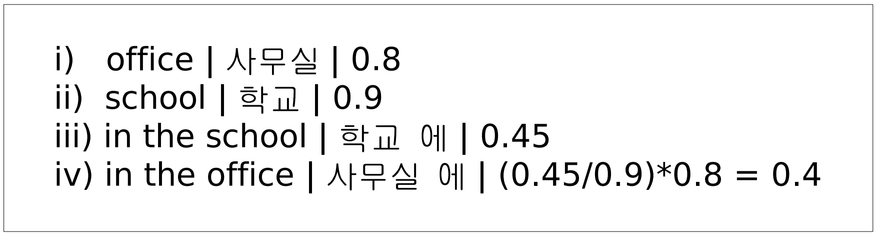
[2] An atomic legal chunk sequence pair is a legal chunk sequence pair that does not include a legal chunk sequence pair.

Korean and English sentence pair. The transliteration of the Korean sentence is “[jeo] [,] [aekjeongpaeneol] [joomoon e] [gwanhae] [jeonhwadeuryeotneundaeyo] [.]” which literally means “[well] [,] [lcd] [order to] [related/about] [am calling] [.] [3]”. The black boxes denote the intersection of Korean to English alignment and English to Korean alignment. The gray boxes are the alignment points that are in the union but not in the intersection. Three of them are added to the final alignment by the *Refined Method* method. After alignment refinement is done, chunk translation sequence pairs are extracted based on the alignment. The rectangular areas with thicker lines denote the extracted chunk translation pairs. The phrase length limit (or the maximum match length in the EBMT system) can control the extraction of pairs.

6.2.2 Chunk fuzzy matching

Hewavitharana et al. (2005) studied translation by similar source sentences. By calculating edit distance, they found similar source sentences. To generate target translation hypothesis, they inserted/replaced/deleted target words that are aligned to the edited source words.

We take a similar approach in our chunk fuzzy matching. Instead of finding similar sentences, we find similar source chunks for unknown chunks in an input sentence.



i) office | 사무실 | 0.8
ii) school | 학교 | 0.9
iii) in the school | 학교 에 | 0.45
iv) in the office | 사무실 에 | (0.45/0.9)*0.8 = 0.4

Figure 6.2: Chunk pair generation

Figure 6.2 shows an example of how we generate a new chunk pair from the chunk pairs extracted by chunk alignment. Suppose that we already learned the chunk translation

[3] The Korean sentence is missing a subject. And there is an error in the Korean sentence chunking. [order to] and [related/about] should be merged into one chunk. However this error was resolved by consistent chunk translation sequence pair extraction.

pairs *i, ii, iii* and now we have a new chunk *iv* to translate. Although we do not have a complete match for the source chunk in *iv*, it is composable using already learned translation pairs and we can modify the target translation chunk accordingly.

We use this approach when there is any chunk that does not have a match for a given source sentence to be translated. When an input sentence is given, we first analyze it into chunks. Then for an unseen input chunk $\mathbf{u}$ from the input sentence, we find a set of similar source chunks S and their translations

$$\begin{aligned} S &= \{(\mathbf{f}_i, \mathbf{e}_i) | \mathbf{f}_i \in V_c, \\ &\quad \mathbf{f}_i \approx \mathbf{u}, \\ &\quad p(\mathbf{e}_i | \mathbf{f}_i) > 0\} \end{aligned}$$

from the training set. These similar source chunks $\mathbf{f}_i$, which belong to the chunk vocabulary V_c of the training set, differ from $\mathbf{u}$ by at most n words. In our experiments, we used $n = 1$ because we wanted to maximize the context similarity. In other words, we prefer source chunks that are different by one word so that the similar chunk $\mathbf{f}_i$ and the unknown input chunk $\mathbf{u}$ share more contexts.

After collecting a similar chunk $\mathbf{f}_i$, we create a template from it by replacing the different source word with a variable. The creation of this template is done on the target side as well by replacing the target word with a variable which is aligned to the different source word (i.e., in the target chunk $\mathbf{e}_i$, we replace the translational equivalent of the different source word with the same variable as the source word variable).

$$(\mathbf{f}_i, \mathbf{e}_i) \rightarrow (\mathbf{f}'_i, \mathbf{e}'_i)$$

Next, for the different source word in $\mathbf{u}$, we use a translation word dictionary to find translational equivalents. With the translation word pairs, we replace the variables in the chunk translation pair templates to get chunk translation pairs for the unknown chunk $\mathbf{u}$.

$$(\mathbf{f}'_i, \mathbf{e}'_i) \rightarrow (\mathbf{u}, \mathbf{e}''_i)$$

The synthesized source chunks from the source chunk templates $\mathbf{f}_i$ are exactly the same as the original unknown input source chunk $\mathbf{u}$. However, the generated target chunks $\mathbf{e}''_i$ are novel, and these may be equally likely or unlikely to appear in the real world. If they are realistic, we have a chance of acquiring a good translation; if not, we should discard them because of their potential to lower the translation quality. To filter out unrealistic generated target chunks, we use a large monolingual language model. We calculate a language model score for each target chunk $\mathbf{e}''_i$ and normalize it by the chunk length which is the number of words in the chunk.

In this way, we can virtually expand our corpus and expect more matches from the corpus at translation time. Furthermore, by generating chunk pairs from existing pairs, we anticipate exploiting the context and reordering that is contained in the existing pairs as well.

For better understanding, we explain this process again with an example. For an unknown source chunk '객실 하나 을' (guest-room one ACCUSATIVE),

- The system first retrieves similar source chunks through substitution and their translations from the consistently aligned chunk pairs. The blue words in Korean are different words in similar chunks and the blue words in English are their translations in the translation chunks.

호텔 하나 을	choose a hotel
편지 하나 을	took a letter
카드 하나 을	a card
방 하나 을	a room

- Next it generalizes them into templates using word alignment links. It replaces the different words with the same variable when they are different from the same word in the unknown chunk.

@1 하나 을	choose a @1
@1 하나 을	took a @1
@1 하나 을	a @1
@1 하나 을	a @1

- And it looks up the different word in the automatically derived dictionary for word replacement. The dictionary entries should be above a threshold.

객실	room	0.3125000

- With translation word pairs (dictionary entries), it replaces the variable and adds the generated chunk pairs to the lattice if the target phrase LM score is above a threshold.

객실 하나 을	choose a room
객실 하나 을	took a room
객실 하나 을	a room
객실 하나 을	a room

- The new chunk pairs are given the feature scores of the corresponding similar chunk pair.

6.2.3 System integration

Figure 6.3 shows how the components are integrated to build a chunk-based system. When an input sentence is given, the system takes following steps in the given order:

1. It finds chunk boundaries for the input sentence using a monolingual chunker.

2. It performs normal surface form matching over the training set for the input sentence.

3. It recognizes chunk matches among all possible matches by using the chunk boundaries found in step 1 and finds chunk translations for them that are already stored in its example-base. The system assigns chunk translations feature scores through the SPA feature scoring functions and an additional feature that indicates these translation are from chunk alignment. Finally, it puts the pairs into a lattice.

4. For the chunks that do not have matches, it tries fuzzy matching. Successfully generated chunk translation pairs are added into the lattice.

5. It performs SPA on the remaining matches to find translations of them.

6. The system uses chunk translations and phrase translations in decoding with a word language model.

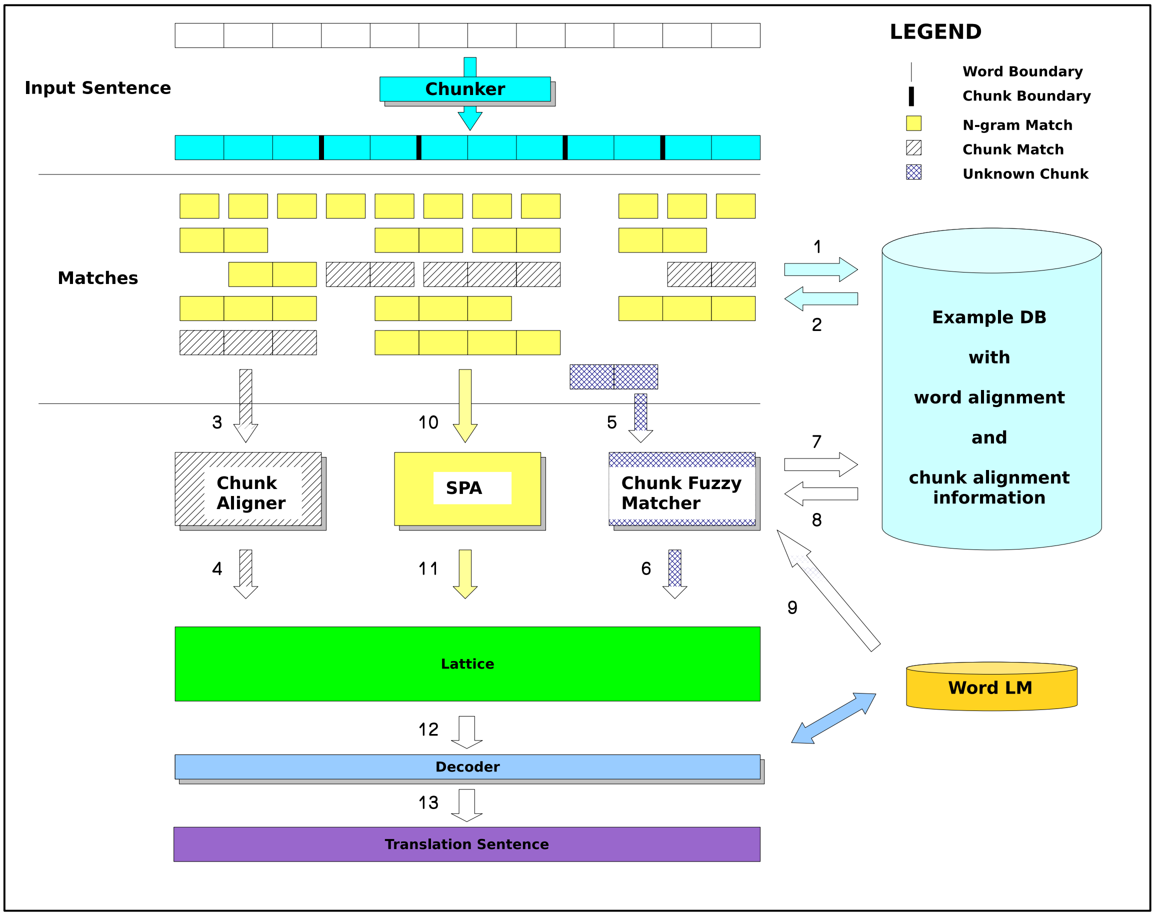


Figure 6.3: System integration

When there is an input sentence, the system first finds chunk boundaries monolingually. We use the same monolingual chunker that is used to find chunk boundaries for the source side of the training data. This chunk boundary information is then used in recognizing chunk matches later.

Next, the system performs the same general surface form matching over the training set for the input sentence that the lexical CMU EBMT system does. The matches found in this include both chunk matches and non-chunk matches.

For the chunk matches, we look up the chunk translation that was built in the chunk alignment stage [4]. Once a chunk translation is found, we put it into a lattice so that our decoder can consider its use in the final translation.

For the unknown chunks (i.e., the chunks for which the system could not find any match in the training set), the system tries fuzzy matching against a chunk translation table which was built during the training time. Then high quality generated chunk pairs are added to the lattice.

For non-chunk matches or chunk matches with no translations, the system invokes SPA to find translation candidates for them and put the translations into the lattice.

After adding all the translations of chunk matches and non-chunk matches, the system loads a chunk label language model and a lexical language model to use in decoding.

In addition to SPA features, we add a feature to the lattice that indicates whether the translation was found by chunk alignment or not. The EBMT system collects some more feature values outside the aligner to be used in decoding.

Finally, the translation with the highest score is chosen as the best translation hypothesis. The score is calculated as a combination of feature values with their weights tuned in a separate tuning process in a log linear model.

6.3 Evaluation

The chunk-based approach is potentially more beneficial for a distant language pair. If we have a very similar language pair in terms of sentence structures and word correspondence, we have very accurate alignment, which gives high quality translation. However, if we have a very distant language pair, it is much harder to align words due to lower sentence structure agreement and word correspondence. Thus the translation quality will be much poorer. But those disagreements are less important in chunk level alignment because sentence structures are much simpler at the chunk level, and source and target chunks have

[4] The extracted chunk translation pairs are annotated in the example database in our implementation.

higher correspondence than source and target words. Therefore if we align chunks in a distant language pair and translate by chunks, we can obtain better translation quality.

To evaluate this chunk-based translation approach, we use Chinese-English and Korean-English which are relatively distant language pairs and French-English which is a close language pair.

Although Chinese is classified as an SVO language like English, it is also very different from English in that it is a topic-prominent language, has aspect and mood particles, and it requires a classifier in counting nouns. It also lacks a lot of correspondents to English function words. Therefore if we translate Chinese to English by chunks, we are likely to have benefit by including English function words that do not have translational equivalents in Chinese in output translations. For example, the translational equivalents of ‘a’, ‘an’ and ‘the’ do not exist in Chinese, and we may expect those to be inserted in translation by chunks.

Korean is also very different from English. Foremost among these differences, it is an SOV language where a verb follows an accusative. It also has case markers that are absent in English, and it lacks some of the English functional words. For example, it does not have articles. Instead it uses numbers for ‘a/an’ and directives for ‘the’ or omits them. In translation into English, some Korean case markers should be removed, and some English articles should be inserted. For example, when we translate ‘sa-moo-sil yi’ into English, which means ‘office NOMINATIVE’, we have to drop ‘yi’ and add an ‘an’ or a ‘the’ in front of ‘office’ depending on the context. Moreover, when there is a correspondent for a case marker, their positions are different. In English, a preposition comes before a head word but its correspondent case marker in Korean follows the head word. For example, in the ‘to the office’ and ‘sa-moo-sil lo’ translation pair, ‘to’ is located in front of its head word ‘office’, but its correspondent ‘lo’ in Korean is located after its head word ‘sa-moo-sil’. Chunk-based translation can be helpful in this case although it does not fully resolve the context recognition problem.

Since we think that this approach is most beneficial for linguistically distant language pairs, we chose the above two languages. However, although they are both distant from English, there is no significant similarity in their sentence structures. So this choice will show not only that our approach is useful for a single distant language pair, but also that it works for distant language pairs in general.

We compare the chunk-based system with the CMU EBMT system with SPA in this evaluation. We measured translation performance differences among:

- **The best SPA**
  This is the best of cSPA-AmX and nSPA-AmX from Chapter 4.

- **cCHUNK-AmX**
  This is a chunk-based system which is a combination of chunk alignment and cSPA-AmX. We added one more feature that indicates whether a phrase translation is by chunk alignment or not.

- **nCHUNK-AmX**
  This is a chunk-based system which is a combination of chunk alignment and nSPA-AmX. We added one more feature that indicates whether a phrase translation is by chunk alignment or not.

We also used Moses to compare with a state-of-the-art system. We use exactly the same data with the same pre-processing in Moses for both training and testing. Moreover, to test the usefulness of the aligned chunks, we extract aligned chunk pairs and add them to a Moses phrase table. We run Moses on both the original phrase table and the chunk-pair-added phrase table.

To compare their performances, we used BLEU as our evaluation metric because it is a widely accepted metric in the machine translation community. We also provide METEOR scores to see if the improvement is consistent across different metrics. The METEOR was set to use stemming and stemmed synonyms to evaluate performance beyond exact match.

For significance test, we used Paired Bootstrap Resampling by Koehn (2004b) with n=1000.

6.3.1 Data

For all the language pairs, we used the same training sets and test sets described in the evaluation in Chapter 4.

Table 6.1 describes Korean-English training data. In Korean, they are an average of 6.5 chunks and 8.9 words long, and in English, they are an average of 6.4 chunks and 9.5 words long. Chunks are 1.4 words and 1.5 words long on average in Korean and English respectively.

	sentences	chunks	words
Korean	28,034	182,549	248,263
English	28,034	178,540	266,583

Table 6.1: Training set for Korean-English

Table 6.2 shows the test sets for Korean-English. On average, the sentences in the Dev set are 6.3 chunks and 8.9 words long. The chunks are an average of 1.4 words long. The Unseen set also has an average of 6.3 chunks and 8.9 words long per sentence with chunks an average of 1.4 words long.

	sentences	chunks	words	number of references
Dev	966	6,071	8,591	1
Unseen	1,170	7,422	10,441	1

Table 6.2: Test sets for Korean-English

Table 6.3 shows the coverage of the training set on the test sets in Korean. We calculate word, chunk, and multi-word chunk coverages. First, we calculate the word coverage to determine the percentage of words that can be translated using a typical word/phrase-based translation system. Second, the chunk coverage is calculated to ascertain what portion can be translated by chunks. Finally, we measured the multi-word chunk coverage because we are most likely to reap benefits by translating chunks which are longer than 1 word to properly deal with word deletion and insertion as previously explained.

	word (%)		chunk (%)		multi-word chunk (%)	
	type	token	type	token	type	token
Dev	82.58	94.53	74.11	86.23	69.80	79.08
Unseen	87.57	96.09	83.12	92.48	81.94	89.26

Table 6.3: Training set coverage for Korean-English

Table 6.4 describes Chinese to English training data. On average, Chinese sentences are 18.1 chunks and 26.8 words long with chunks composed of 1.5 words. And English sentences are an average of 18 chunks and 33.9 words long with chunks composed of 1.8 words.

	sentences	chunks	words
Chinese	341,636	6,177,252	9,155,903
English	341,636	6,419,184	11,571,835

Table 6.4: Training set for Chinese-English

Table 6.5 shows Chinese-English test sets. On average, the sentences in the Dev set are 17.5 chunks and 46.7 words long. The chunks are an average of 2.67 words long. The Unseen set also has an average of 17.0 chunks and 45.9 words long per sentence with chunks an average of 2.67 words long.

	sentences	chunks	words	number of references
Dev	919	16,083	42,946	4
Unseen	691	11,786	31,708	4

Table 6.5: Test sets for Chinese-English

Table 6.6 shows the coverage of the training set on the test sets in Chinese. Word, chunk, and multi-word chunk coverage are reported as in Table 6.3 for Korean. The multi-word chunk coverage is much lower for Chinese compared to Korean (34.09% vs 69.80%

on chunk type for the Dev sets). On average the covered chunks have 2.22 words and the uncovered chunks have 3.66 words in the Chinese Dev set. This means our Chinese chunker tends to find long chunks, which leads to low multi-word chunk coverage. This may be improved by looking at the chunked data to change chunking strategies for finding shorter chunks.

	word (%)		chunk (%)		multi-word chunk (%)	
	type	token	type	token	type	token
Dev	88.00	96.24	59.45	81.83	34.09	38.71
Unseen	89.09	95.88	63.19	82.50	38.70	43.63

Table 6.6: Training set coverage for Chinese-English

Table 6.7 describes French to English training data. On average, French sentences are 17.3 chunks and 30.5 words long with chunks composed of 1.8 words. English sentences are an average of 16.0 chunks and 28.0 words long with chunks composed of 1.7 words long on average.

	sentences	chunks	words
French	300,000	9,143,101	5,191,557
English	300,000	4,814,544	8,402,980

Table 6.7: Training set for French-English

Table 6.8 shows French-English test sets. On average, the sentences in the Dev set are 18.6 chunks and 32.2 words long and the chunks are an average of 1.73 words long. The Unseen set also has an average of 16.7 chunks and 30.0 words per sentence with chunks an average of 1.74 words long.

Table 6.9 shows the coverage of the training set on the test sets in French. The coverages for the three types are comparable to those in the Korean test sets.

In addition to the chunk-based system evaluation, we also investigate if the aligned chunk pairs can help an SMT system for which we chose Moses. We add our chunk trans-

	sentences	chunks	words	number of references
Dev	285	5,302	9,174	1
Unseen	2,007	33,482	58,168	1

Table 6.8: Test sets for French-English

	word (%)		chunk (%)		multi-word chunk (%)	
	type	token	type	token	type	token
Dev	97.93	99.48	85.00	91.70	79.95	83.16
Unseen	91.38	98.19	69.34	85.74	62.45	73.41

Table 6.9: Training set coverage for French-English

lation pairs to the extracted phrase pairs found by Moses. For that, we first pause the Moses training process after step 5 and add our chunk translation pairs to the intermediate data (extracted phrase pairs). Then we resume the training process so that Moses can assign feature scores to the chunk translation pairs as well. Finally, we execute the Moses decoder on the phrase table generated by the above method.

6.3.2 Results and analysis

Phrase Aligner	Dev		Unseen		# SPA alts on Dev
	BLEU	METEOR	BLEU	METEOR	
cSPA-Am5	0.2468	0.4687	0.2552	0.4603	5
cCHUNK-Am7	**0.2480**	0.4709	**0.2565**	0.4662	7
nCHUNK-Am3	0.2456	0.4654	0.2561	0.4618	1.234
Moses	0.2203	0.4323	0.2353	0.4362	N/A

Table 6.10: Korean to English translation performance (BLEU/METEOR)

Table 6.10 shows performance comparisons between our baseline system (cSPA-Am5) and the new Chunk-Based EBMT, cCHUNK-Am7. For both the development set and unseen set, cCHUNK-Am7 performs slightly better than the baseline system. Our significance test indicates that the improvements are not significant with $p = 0.378$ and $p = 0.318$ respectively.

We also compared the Korean-English results with the performance of the Moses system to see how closely the EBMT performs to one of the well known state-of-the-art systems. To make them comparable, we trained the Moses system on the same data and set the decoding parameters of both systems comparably. i.e., we used the same values for the corresponding parameters of the Moses system and the EBMT system. For example, we used the same value for “distortion-limit” in the Moses system and its corresponding parameter “reorder-window” in the EBMT system. In Table 6.10, the EBMT system outperforms the Moses system for both Dev and Unseen with $p < 0.0001$.

Aligner		SPC	ASPL	ATPL	SNTPC
cSPA-Am5	Total	4,502	1.49	1.64	0
nSPA-Am4	Total	4,447	1.49	1.61	263
cCHUNK-Am7	Chunk	1616	1.38	1.45	0
	Chunk-{P}	922	1.66	1.80	0
	Non-chunk	2932	1.53	1.64	0
	Total	4548	1.48	1.57	0
nCHUNK-Am3	Chunk	1215	1.45	1.42	0
	Chunk-{P}	764	1.72	1.67	0
	Non-chunk	2848	1.72	1.74	78
	Total	4063	1.64	1.65	78

Table 6.11: Korean-English selected phrase statistics in decoding on Dev

Table 6.11 shows selected phrase statistics in decoding. In this table, **SPC** denotes Selected Phrase Count, **ASPL** denotes Average Source Phrase Length, **ATPL** denotes Average Target Phrase Length, and **SNTPC** denotes Selected Non-contiguous Target Phrase Count. *Chunk* - $\{P\}$ denotes that we did not count punctuation translations by chunk alignment. *Chunk* denotes that the translation phrase pairs are from chunk alignment, *SPA* denotes that the translation phrase pairs are from SPA alignment, and *Total* denotes all the translation pairs. Since chunk translations include a lot of punctuation translations [5] which can also be provided by SPA algorithms, we counted chunk translations excluding punctuation translations in *Chunk* - $\{P\}$. When we compare cCHUNK-Am7 and nCHUNK-Am3 systems, we note that cCHUNK-Am7 selected more chunk translations than nCHUNK-Am3, and the number of source words covered by chunk phrases is larger in cCHUNK-Am7. This shows the use of cSPA leads the combined system to select more chunk translations and achieve better translation performance with them.

Phrase Aligner	Dev		Unseen		# SPA alts on Dev
	BLEU	METEOR	BLEU	METEOR	
cSPA-Am5	0.2423	0.5242	0.1996	0.4774	5
cCHUNK-Am5	0.2467	0.5196	0.2020	0.4795	5
nCHUNK-Am5	**0.2541**	0.5302	**0.2059**	0.4885	1.295
Moses	0.2593	0.5365	0.2070	0.4974	N/A

Table 6.12: Chinese to English translation performance (BLEU/METEOR)

Table 6.12 shows the BLEU/METEOR scores for the Chinese test sets. For both sets, the chunk-based system, nCHUNK-Am5 demonstrates significant improvements over cSPA-Am5 with $p < 0.0001$ and $p = 0.027$ respectively. Of note, nCHUNK-Am5 is better than cCHUNK-Am5, which is opposite to what we observed in Chapter 4. In Chapter 4, contiguous SPA was better than non-contiguous SPA, but in this experiment, when combined with chunk translation, non-contiguous SPA is better than contiguous SPA.

Table 6.13 shows selected phrase statistics in decoding for Chinese-English translation. In this table, **SPC** denotes Selected Phrase Count, **ASPL** denotes Average Source Phrase Length, **ATPL** denotes Average Target Phrase Length, and **SNTPC** denotes Selected Non-contiguous Target Phrase Count. *Chunk* - $\{P\}$ denotes that we did not count punctuation

[5] Forms of punctuation are also chunks according to our definition and they are aligned accurately by the chunk aligner. This alignment information is stored in the example-bases of the EBMT system. For this reason, when an input sentence has punctuations, they are translated by chunk alignment.

translations by chunk alignment. When we compare the SPA systems (cSPA-Am5 and nSPA-Am5) with CHUNK systems(cCHUNK-Am5 and nCHUNK-Am5), we note that the chunk-based systems selected longer phrases on average. Furthermore, when we compare cCHUNK-Am5 and nCHUNK-Am5 systems, we note that nCHUNK-Am5 selected more chunk translations than cCHUNK-Am5 and that the average length of the selected chunk phrases is much longer in nCHUNK-Am5. This shows the use of nSPA leads the combined system to select more chunk translations and achieves better translation performance. Also of interest, the length ratio of target phrases over source phrases is much larger by chunk translations than by non-chunk translations. For example, in nCHUNK-Am5, the ratios are 1.23 and 1.09 by chunk translations and non-chunk translations respectively. Because forms of punctuation were translated by chunk alignment, we compared *Chunk* -$\{P\}$ and *Non-chunk*.

For the Dev set, Moses performs the best, but for the Unseen set, nCHUNK-Am5 performs as well as the Moses system.

Aligner		SPC	ASPL	ATPL	SNTPC
cSPA-Am5	Total	15,850	1.41	1.57	0
nSPA-Am5	Total	16,005	1.37	1.52	270
cCHUNK-Am5	Chunk	2,763	1.15	1.30	0
	Chunk-{P}	1,423	1.29	1.58	0
	Non-chunk	12,831	1.50	1.64	0
	Total	15,569	1.44	1.58	62
nCHUNK-Am5	Chunk	3,870	1.23	1.44	0
	Chunk-{P}	2,455	1.37	1.69	0
	Non-chunk	11,752	1.48	1.61	353
	Total	15,607	1.42	1.57	353

Table 6.13: Chinese-English selected phrase statistics in decoding on Dev

Table 6.14 shows translation results for French-English. Although both cSPA-Am5 and nCHUNK-Am4 perform worse than Moses, nCHUNK-Am4 demonstrates improvement

Phrase Aligner	Dev		Unseen		# SPA alts on Dev
	BLEU	METEOR	BLEU	METEOR	
cSPA-Am5	0.2409	0.5390	0.1924	0.5368	5
cCHUNK-Am5	0.2451	0.5351	0.1925	0.5224	5
nCHUNK-Am4	**0.2506**	0.5506	**0.2040**	0.5545	1.303
Moses	0.2516	0.5527	0.2102	0.5511	N/A

Table 6.14: French to English translation performance (BLEU/METEOR)

over cSPA-Am5 on both Dev and Unseen with $p = 0.001$ and $p < 0.0001$ respectively.

Aligner		SPC	ASPL	ATPL	SNTPC
cSPA-Am5	Total	3,700	2.49	2.24	0
nSPA-Am5	Total	4,214	2.16	1.95	147
cCHUNK-Am5	Chunk	477	1.07	1.00	0
	Chunk-$\{$P$\}$	285	1.12	1.00	0
	Non-chunk	4,730	1.78	1.58	0
	Total	5,201	1.71	1.53	0
nCHUNK-Am4	Chunk	474	2.01	1.87	0
	Chunk-$\{$P$\}$	373	2.28	2.11	0
	Non-chunk	3,433	2.41	2.15	105
	Total	3,903	2.36	2.11	105

Table 6.15: French-English selected phrase statistics in decoding on Dev

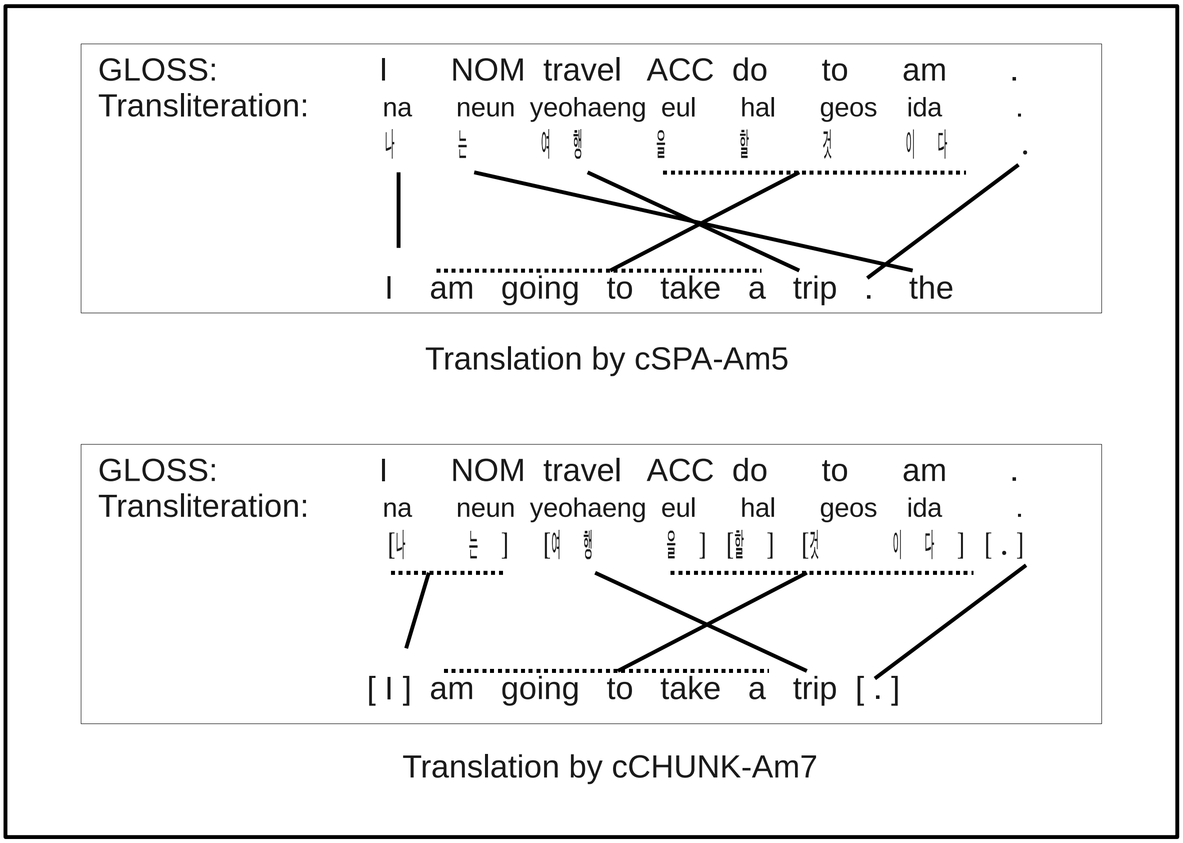
Table 6.15 shows selected phrase statistics in decoding for French-English translation. In this table, **SPC** denotes Selected Phrase Count, **ASPL** denotes Average Source Phrase Length, **ATPL** denotes Average Target Phrase Length, and **SNTPC** denotes Selected Non-contiguous Target Phrase Count. *Chunk* - $\{P\}$ denotes that we did not count punctuation translations by chunk alignment. When we compare the SPA systems (cSPA-Am5 and nSPA-Am5) with CHUNK systems(cCHUNK-Am5 and nCHUNK-Am5), we note that the chunk-based systems selected longer phrases on average. And when we compare cCHUNK-Am5 and nCHUNK-Am4 systems, for *Chunk* - $\{P\}$, we note that nCHUNK-Am5 selected more chunk translations than cCHUNK-Am5 and that the average length of the selected chunk phrases is much longer in nCHUNK-Am4. This shows that the use of nSPA leads the combined system to select more chunk translations and achieves better translation performance.

Table 6.16 reports the portions of phrasal translations that are chunk translations in decoding. The portions of chunk translations are about 35.5%, 17.7%, and 9.2% in the Korean to English, Chinese to English, and French to English the Dev set translation tasks respectively. We notice that the portion is the highest in Korean to English translation, and the lowest in French to English translation. We think this is because of the ratio of the words that do not have translational equivalents. In other words, chunk translation is more critical to Korean to English translation in order to properly deal with the multitude of words which do not have translational equivalents while chunk translation is less important in French to English translation due to better word alignment accuracy.

Language Pair	Set	Phrasal Translations	Chunk Translations	%
Kr-En	Dev	4,548	1,616	35.5
	Unseen	5,333	1,472	32.7
Cn-En	Dev	15,569	2,763	17.7
	Unseen	11,343	2,288	20.2
Fr-En	Dev	5,201	477	9.2
	Unseen	32,288	3,641	11.6

Table 6.16: Chunk translations used in decoding

Figure 6.4 shows an excellent actual translation example for which chunk translation was beneficial. In the baseline system, the Korean nominative case marker was translated to ‘the’ in English although it should be dropped or translated to ‘I’ together with the Korean word ‘na’. But in the chunk-based system, the Korean chunk consisting of ‘na’ and the nominative case marker was translated into the English chunk ‘I’ correctly.


GLOSS: I NOM travel ACC do to am .
Transliteration: na neun yeohaeng eul hal geos ida .
나 는 여행 을 할 것 이다 .
I am going to take a trip . the
GLOSS: I NOM travel ACC do to am .
Transliteration: na neun yeohaeng eul hal geos ida .
[나 는] [여행 을] [할] [것 이다] [.]
[I] am going to take a trip [.]

Translation by cSPA-Am5

Translation by cCHUNK-Am7

Figure 6.4: Translation example by cCHUNK-Am7

Chunk pair generation

As shown in Tables 6.3, 6.6, and 6.9, there are many unknown multi-word chunks. We investigate if chunk translation pair generation can help translation.

For the Korean Dev set, we have 830 OOV chunks and 193 chunks out of them consist of three or more words [6]. From the extracted chunk translation pairs, we find similar chunks [7] for 135 chunks out of the 193 chunks with a phrase (chunk) translation probability threshold of 0.1. Sixty-seven in the 135 different words out of the 135 chunks are identified to have a translation word in a word translation dictionary with a threshold of word

[6] We use the OOV chunks that are at least three words long to maximize context similarity.

[7] We define a similar chunk to be different by one word from the an OOV chunk.

translation probability 0.2, which was empirically chosen with regard to the dictionary. Finally, by applying a language model probability threshold of 0.0001 for the generated target side, we filter out the generated chunks that are not likely to appear in the real world and obtain chunk translation pairs for the OOVs in 21 out of the 966 sentences.

We also assess the approach for Chinese to English translation. We apply the same threshold values and generate chunk translation pairs for 49 out of 230 sentences.

Table 6.17 records the development scores in BLEU for both language pairs. Although we cannot claim statistically significant improvements with such a small number of sentences, we think it may be meaningful that we had improvement for both language pair cases, particularly because our results indicate a possible improvement in a case where we have many unknown chunks.

Lang. Piar	EBMT(Chunk)	EBMT(Chunk) w/ Generation
Korean-English (21 sentences)	0.1696	**0.1780**
Chinese-English (49 sentences)	0.2755	**0.2817**

Table 6.17: Chunk translation pair generation results

Chunk translation pairs for Moses

We investigated whether our chunk translation pairs can help a Phrased-Based SMT system in Korean-English and Chinese-English translation. We chose Moses as the PBSMT system since it works on phrase pairs and is considered a state-of-the-art system. To combine chunk translation pairs and Moses phrase pairs, we appended chunk pairs to extracted Moses phrase pairs before the phrase score calculation step so that they are assigned Moses feature scores in the same way. As in Table 6.18, the improvements in Korean-English are statistically significant for both *Dev* and *Unseen* sets. In the Chinese-English translation, the improvement for *Unseen* is significant although it is difficult to demonstrate the significance of the improvement for *Dev*.

We added all the chunk translation pairs to Moses in Korean-English, but in Chinese-

Lang. Pair	System	Dev		Unseen	
		BLEU	METEOR	BLEU	METEOR
Kr-En	Moses	0.2203	0.4323	0.2353	0.4362
	Moses w/ chunks	0.2282	0.4459	**0.2408**	0.4455
Cn-En	Moses	0.2593	0.5365	0.1938	0.4930
	Moses w/ chunks	0.2605	0.5406	**0.2051**	0.5048

Table 6.18: Improvements in Moses with chunk translation pairs

English, the results were the best when we added chunk translation pairs with frequency three and higher.

This means that we should exercise caution when filtering out low quality chunk translation pairs depending on the training set size. Table 6.19 shows the phrase table size changes after we added the chunk translation pairs to the Moses phrase tables. In this table, *#PP* denotes number of phrase pairs, *#SP* denotes number of unique source phrases and *#WT* denotes number of word types. For both language pairs, the method added new source phrases and words and the added amount is relatively smaller in the Chinese-English case. We think this is because the training set is much larger than that of the Korean-English case. A larger training set led to better word alignments and consequently helped in building a better phrase table. In other words, the Korean-English training set was very small and Moses built a low quality phrase table because word alignments were not accurate. So, even less accurate chunk translation pairs were helpful in this case.

Overall, the results show that we can improve Moses by adding carefully chosen chunk translation pairs.

Summary

To summarize, firstly, the chunk-based system shows significant improvements over our baseline EBMT system that uses SPA for phrasal alignments for all of the three language pairs of Korean-English, Chinese-English and French-English.

Lang. Pair	System	Dev			Unseen		
		#PP	#SP	# WT	# PP	#SP	# WT
Kr-En	Moses	11,040	4,140	1,683	12,492	4,918	1,926
	Moses w/ chunks	12,492	4,598	1,781	15,819	5,392	2,016
Cn-En	Moses	330,951	15,599	4,226	321,083	13,330	3,562
	Moses w/ chunks	337,546	15,619	42,33	327,665	13,339	3,563

Table 6.19: Moses phrase table size

Secondly, the chunk translation pair generation helped the small amount of input data. However, because the number of affected sentences is too small, we cannot sufficiently demonstrate the statistical significance of the improvement. But we suspect the possibility of potential improvement where we have a significant portion of unknown multi-word chunks.

Finally, chunk translation pairs identified by our chunk alignment algorithm helped a statistical machine translation system, Moses in this experiment. By using carefully chosen additional chunk translation pairs, we were able to improve Moses for Chinese-English and Korean-English translation.

6.3.3 The effect of ideal chunking

Table 6.20 shows some example sentences chunked by different chunkers [8]. The Korean sentence were chunked by a human to show an ideal chunking and are followed by glosses. The corresponding English sentences were chunked by two different chunkers E_A and E_B. When we ran the two English chunkers on the Korean-English training set, 10,741 English sentences were chunked differently by them which is about 38.3%. Note that this difference is on short sentences and if we apply them on longer sentences, we will have

[8] **NOM** denotes nominative, **QUA** denotes quantative, **ACCU** denotes accusative, **ASK** denotes a question case marker and **TOPIC** denotes a case marker for topic. Numbers in parenthesis in the glosses mean that the word corresponds to as many words in Korean as the number.

even more differently chunked sentences.

Chunking errors: The chunking is different due to mainly wrong chunking by a chunker. In our examples, the chunker E_A produces more wrong chunks compared to the chunker E_B. For example, the chunker E_A malfunctioned by combining two chunks in “at+the+hotel+last+night” and “a+plane+slides” in sentence 98 and 113 respectively, not combining multiple words into one chunk in “a one-way”, “a sightseeing train” and “about how much” in sentence 107, 114 and 127 respectively Although the chunker E_B chunked “to+new york” and “about how+much” wrong in sentence 118 and 127 respectively, it produces fewer errors in our examples. Chunking on their corresponding Korean sentences clearly shows that these errors are not desirable because they hurt correspondence [9].

Bilingual chunking: In our example, the chunking errors shown above may be overcome to some degree if they are provided with a good algorithm which involves phrase detection and takes into account good chunking on the other language. For example, “to new york” and “about how much” can be chunked correctly by looking at the corresponding Korean sentences.

Structural problems: Sentence 123 shows a structural problem in chunking caused by the structural difference of the two languages. The Korean chunks “delayed_or route+ACCU change+do+may(2)” is only meaningful when it is aligned to the English chunks “may+be+delayed+or+forced to+re-route”. There is no one to one mapping between those two parts. In this case, we have to use the Korean chunks as one unit in translation, but this in turn causes lower coverage.

In these observations, the chunker E_B gives lower chunking errors and better correspondence to Korean chunks, given that the Korean side is chunked ideally. It may also be possible that having bilingual chunking adjustments in chunking may reduce chunking errors. From these examples, it is not difficult for one to see that erroneous chunk-

[9] For sentence 107, combining “a one-way” into a chunk hurts local chunk correspondence because the corresponding Korean words are chunked into two chunks, “one_way+ticket” and “1-QUA”. However, because the English word “a” is not a good translation of the Korean chunk “1-QUA”, splitting “a” and “one-way” in the English sentence is not good chunking.

ing hurts correspondence and, consequently, translation. In our experiments for Korean-English translation, we used the chunker E_A which yields more chunking errors because we were not aware of the chunker E_B when we started the experiments. It may be possible that we can achieve better translation performance with the chunker E_B as our English chunker.

ID	Chunker	Chunked sentence pair
98	H	어젯밤+에 그+호텔+에 불+이 났었다 .
		last_night-in that+hotel+at fire+NOM broke_out .
	E_A	a+fire broke out at+the+hotel+last+night .
	E_B	a+fire broke out at+the+hotel last+night .
107	H	서울 편도+표 1+장 주세요 .
		seoul one_way+ticket 1+QUA give_me_please .
	E_A	a one-way to+seoul , please .
	E_B	a+one-way to+seoul , please .
113	H	비행기+이 물+위+을 활주한다 .
		plane+NOM water+over+ACCU slides .
	E_A	a+plane+slides over+the+water .
	E_B	a+plane slides over+the+water .
114	H	관광열차요 ?
		sightseeing_train_ASK ?
	E_A	a sightseeing train ?
	E_B	a+sightseeing+train ?
118	H	뉴욕+까지 1+장 주십시오 .
		new_york+to 1+QUA give_me_please .
	E_A	a+ticket to+new+york , please .
	E_B	a+ticket to+new york , please .
123	H	여행+은 여러+가지+이유+으로 지체되거나 여정+을 바꿔야+할+수도+있다 .
		trip+NOM various(2)+reason+for delayed_or route+ACCU change+do+may(2) .
	E_A	a+trip may+be+delayed or forced to+re-route for+various+reasons .
	E_B	a+trip may+be+delayed+or+forced to+re-route for+various+reasons .
127	H	그+호텔+까지+는 요금+이 대략+얼마쯤 됩니까 ?
		that+hotel+to+TOPIC fare+NOM about+how_much is ?
	E_A	about how much is the+fare to+the+hotel ?
	E_B	about how+much is the+fare to+the+hotel ?

Table 6.20: Different chunking for Korean-English

Chapter 7

Conclusions

7.1 Conclusions

This work contributes significantly to the field of corpus-based machine translation.

Firstly, SPA improved the translation quality of the CMU EBMT system. Before this work, the CMU EBMT system used a heuristic phrasal aligner, which employed binary correspondence between source words and target words to determine a target translation phrase given a source phrase. It used all the sub-phrases of the longest possible target phrase that completely include the shortest possible target phrase as candidates based on the binary correspondence and returned the one having the highest heuristic score as a translation. Cognizant of the recent strides in the SMT field, we wanted to use a more sophisticated score calculation method instead of the heuristic one. Our new phrasal aligner SPA gave us statistically significant improvements in translation quality. In our small French-English translation experiments, it gave us 20∼35% improvements in BLUE score.

Secondly, the state-of-the-art external word alignment helped SPA. In our experiments we used Moses word alignment as external word alignment and it helped the SPA in two ways. First, it helped SPA in determining a target range from which SPA draws target translation candidates. For Korean-English and Chinese-English which are distant language pairs, SPA performed better in translation with this target range than a proportionally determined target range which assumes that the source and target languages have the same word orders. Secondly, the external word alignment itself was a good translation candidate. When we made SPA return the external word alignment as the best target candidate phrase along with other derived target candidate phrases, the EBMT system performed significantly better for all three language pairs.

Thirdly, non-contiguous SPA (nSPA) did not perform better than SPA (cSPA) except for the French-English Unseen set. Moreover, for Chinese-English, it performed significantly worse. The nSPA returned less than 1.5 translation candidates on average which gave lower coverage. However, nSPA is more than 10 times faster in translation time which includes both alignment and decoding time because its search space is much smaller by investigating only includable/removable words. Importantly, when there are a lot of includable/removable words, the system can become very slow because it investigates 2^{i+r} candidates. This slowing did not occur in our experiments which used a setting of maximum source phrase length being 7.

Fourthly, chunk alignment was better when it used both chunk pair statistics and word pair statistics than when it used only one of the two. After investigating SPA we moved to exploring the benefits of using chunks as basic translation units. To investigate chunk translation in the EBMT system, we first investigated chunk alignment. We developed a chunk alignment algorithm that boosts a chunk pair alignment when included source and target words are aligned (*Word-boost*). This was better than when we simply aligned a chunk pair when there is a word alignment link (*Word-map*) and when we regarded a chunk as a unit in alignment by concatenating all the words in a chunk and aligned them (*Baseline*). Then we recognized consistently aligned chunk sequence pairs to use in translation. When we restricted alignment evaluation to consistently aligned chunk sequence pairs, *Word-boost* was better than SPA phrasal aligner. However, because this was worse than the SPA aligner with top-10 candidates (SPA-(Top-10)), SPA can potentially perform better than *Word-boost*.

Fifthly, chunk-based translation improved translation quality when used with SPA. When we combined *Word-boost* with SPA or non-contiguous SPA, it performed better than SPA and nSPA. The best performing variants had translation quality improvements over the best performing SPA and nSPA. The improvements were significant for Chinese-English and French-English, but slight for Korean-English. Of note, chunk alignment worked better when it was combined with nSPA than cSPA although nSPA performed worse than cSPA. Our analysis showed that more longer chunk translations were selected when we combined it with nSPA, which is not surprising because nSPA is more likely to return non-contiguous alignment for longer source phrases which have lower language model scores and thus are hard to be selected by the decoder that does not interlock non-contiguous target phrases.

Sixthly, chunk alignment can provide useful chunk translation pairs to PBSMT. We added consistent chunk translation pairs to a Moses phrase table. Moses performed better when we added the chunk pairs. However, we had to apply a careful filtering mechanism to discern convincing translation pairs and include only them.

Finally, our goal was to attain a 5% relative improvement and we almost achieved it. Table 7.1 shows our achievement. For the baseline system we compare with, we picked *cSPA-m1* which is the worst performing SPA variant because we did not have the performance results for the original heuristic aligner for the latest test sets we used. Because *cSPA-m1* is better than the heuristic aligner, our achievements will be even higher against the heuristic aligner. As the best performing chunk-based system (*CHUNK*), we used *cCHUNK-Am7* for Korean-English, *nCHUNK-Am5* for Chinese-English and *nCHUNK-Am4* for French-English. Our achievements are huge for Korean-English and Chinese-English with 11.06% and 27.05% improvements in BLEU. Interestingly, the achievement is larger for BLEU than METEOR. This is because BLEU tends to obtain a higher BLEU score by having higher precision compared to METEOR which weights 9 times more on recall.

7.2 Future work

The following topics merit further investigation.

Firstly, more features in SPA can be developed for the possibility of improving the

Lang. Pair	BLEU			METEOR		
	cSPA-m1	CHUNK	Imp.	cSPA-m1	CHUNK	Imp.
Korean-English	0.2231	0.2480	+11.16%	0.4400	0.4709	+7.02%
Chinese-English	0.2000	0.2541	+27.05%	0.4787	0.5302	+10.76%
French-English	0.2378	0.2506	+5.38%	0.5384	0.5506	+2.26%

Table 7.1: Improvements achieved

system. Specifically we would investigate word collocation score. For that we would learn word collocation scores and use them for source phrases and target phrases. Given a source phrase and a target translation candidate phrase, if their average collocation scores are very different, they are less likely to be a good translation pair. In this case, we assumed that, in a good translation pair, average source word relationship and average target word relationship are similar and we can use the word collocation scores to measure the relationship. Orliac and Dillinger (2003) extracted collocations based on rules using grammatical features and semantic contexts and Liu et al. (2010) learned collocation scores on word tokens tweaking IBM models. In our case, we can deploy Liu et al. (2010)s method to learn word collocations because it does not require additional linguistic information. Given a source phrase $\mathbf{f} = f_{i+1}^{i+k} = f_{i+1}, ..., f_{i+k}$, we can calculate a collocation score $CL(\mathbf{f})$ as following:

$$CL(\mathbf{f}) = \frac{\sum_{(f_m, f_n) \in P} Collocation\_score(f_m, f_n)}{|P|} \tag{7.1}$$

where $P = \{(f_m, f_n) | i + 1 \leq m < n \leq i + k\}$

Secondly, using word links directly in *Word-boost* would also be of interest. In our work, we calculated word translation probability from the word alignment and used it in the formula 5.5. This time, in addition to boosting chunk mapping counts by word translation probability, we could boost chunk mapping count again by the average of word link score. For example, we can assign a value of 1 to a linked word pair and a value of 0.5 to an unlinked word pair and calculate average word link score in a chunk pair.

Thirdly, detecting/filtering out noisy chunk translation pairs in the EBMT system could be beneficial. We observed that filtering out noisy/less-convincing pairs is helpful when we added chunk translation pairs to the Moses system. For the same reason, we think this will be helpful for the EBMT system as well.

Fourthly, we have only initial results for fuzzy chunk matching. The data set we used was very small, and there were not many generated chunks for the set because we used only substitution in similarity calculation. If we use word insertion/deletion as in Hewavitharana et al. (2005)'s work, we could generate more chunks. Also by adjusting thresholds for phrase translation score, word translation probability, and language score, we could see a different result.

Finally, we could use METEOR as our tuning objective function. As it turned out that METEOR is a better objective function than BLEU for 1 reference sets for tuning by He and Way (2009) and our Korean-English and French-English test sets have 1 reference translation, it will be of interest to tune our parameters for METEOR and see if the improvements are consistent.

Appendix A

Korean to English Translation Examples

In this appendix, we compared some translation examples from *EBMT(SPA)* and *EBMT(Chunk)*. Chunks in the source sentences are wrapped with brackets. Note that there are errors in this automatic chunking.

In the example of Table A.1, '은 올 여름' was translated to 'the sea this summer' in *EBMT(SPA)* while '그녀 은' and '올 여름' were translated to 'she' and 'this summer'. Note that the Korean chunker split '올' and '여름' mistakenly, the chunk aligner made them to be translated together.

Source	[그녀 은] [올] [여름] [유럽 을] [여행했다] [.]
Gloss	[she NOMINATIVE] [this] [summer] [europe ACCUSATIVE] [traveled] [.]
EBMT(SPA)	she traveled through europe . the sea this summer
EBMT(Chunk)	she traveled through europe this summer .
Reference	she traveled in europe this summer .

Table A.1: Translation Example

In the example of Table A.2, '은' was erroneously translated to 'they' by *EBMT(SPA)* while '그녀 은' was translated to 'she' by *EBMT(Chunk)*.

In the example of Table A.3, '이' was translated to 'been' by *EBMT(SPA)* while '무슨 일 이' was translated to 'what' by *EBMT(Chunk)*.

Source	[그녀 은] [어디 에] [앉아야] [할지 모른다] [.]
Gloss	[she NOMINATIVE] [where at] [sit-down] [to do-not-know] [.]
EBMT(SPA)	they do not know where to sit down . she
EBMT(Chunk)	she does not know where to sit down .
Reference	she does not know where to sit down .

Table A.2: Translation Example

Source	[왜] [,] [무슨 일 이] [있었는데] [?]
Gloss	[why] [,] [what matter NOMINATIVE] [happened-QUESTION] [?]
EBMT(SPA)	why , what happened ? been
EBMT(Chunk)	why , what happened ?
Reference	why , what happened ?

Table A.3: Translation Example

In the example of Table A.4, '은' was translated to 'i' by *EBMT(SPA)* while '그 은' was translated to 'he' by *EBMT(Chunk)*. '진지하게' is an unknown token.

Source	[그 은] [정말] [진지하게] [편지 을] [썼다] [.]
Gloss	[he NOMINATIVE] [very] [seriously] [letter ACCUSATIVE] [wrote] [.]
EBMT(SPA)	i wrote a letter . he is a real
EBMT(Chunk)	he wrote a letter . really
Reference	he wrote a letter in all seriousness .

Table A.4: Translation Example

In the example of Table A.5, although the translation by *EBMT(SPA)* is good and closer to the reference, the translation by *EBMT(Chunk)* is also legitimate.

In the example of Table A.6, although both translations sound fluent except the second 'it', the translation by *EBMT(Chunk)* makes more sense.

In the example of Table A.7, although both systems have 0 sentence level BLEU scores, *EBMT(Chunk)* has more possibility for improvement by translating '수 있어요' to 'can i have'. In this case, chunking for the Korean sentence is not good but consistent chunk

Source	[이번 주말 에] [객실 하나 을] [예약하고] [싶습니다] [.]
Gloss	[this weekend on] [room one ACCUSATIVE] [reserve] [want-to] [.]
EBMT(SPA)	i ’d like to make a reservation for a room on weekend .
EBMT(Chunk)	i ’d like to make a reservation for a single room at the end of this week .
Reference	i ’d like to book a room for this weekend please .

Table A.5: Translation Example

Source	[집 에서] [이 곳 까지] [오는데] [얼마나] [걸리나요] [?]
Gloss	[home from] [this place to] [to-come] [how-long] [take-QUESTION] [?]
EBMT(SPA)	how long does it usually take it from here to the house ?
EBMT(Chunk)	how long does it usually take it to this place from my home ?
Reference	how long does it take to get here from your home ?

Table A.6: Translation Example

alignment overcame it to some degree.

Source	[오늘] [나중 에] [거기] [갈] [수] [있게] [예약] [좀] [할] [수] [있어요] [?]
Gloss	[today] [later at] [there] [go] [to] [to-be-able] [reservation] [please] [make] [to] [be-able-Q
EBMT(SPA)	today there later reservation , please ? can you be able to
EBMT(Chunk)	be able to get there on today can i have a reservation , please ?
Reference	can i make an appointment for later today ?

Table A.7: Translation Example

In the example of Table A.8, *EBMT(SPA)* has an alignment error to include ‘operating’ while *EBMT(Chunk)* could not deal with ‘is’ and ‘’m’ properly when ‘he is’ was chosen as a translation of ‘그 은’ by the chunk aligner erroneously.

In the example of Table A.9, *EBMT(SPA)* erroneously translated ‘은’ to ’the’ while *EBMT(Chunk)* took it as a part of ‘나 은’ and translated into ‘i’.

In the example of Table A.10, ‘정거장 에’ was better translated to ’at the station’ by *EBMT(Chunk)*.

Source	[그 은] [이] [지역] [지리 에] [익숙하지] [않아요] [.]
Gloss	[he NOMINATIVE] [this] [area] [ways at] [good] [be-not] [.]
EBMT(SPA)	he is not used to operating this area .
EBMT(Chunk)	he is not ’m used to this area .
Reference	he is not familiar with this area .

Table A.8: Translation Example

Source	[나 은] [여행 을] [할] [것 이다] [.]
Gloss	[i NOMINATIVE] [trip ACCUSATIVE] [do] [to be] [.]
EBMT(SPA)	i am going to take a trip . the
EBMT(Chunk)	i am going to take a trip .
Reference	i ’m going to make a journey .

Table A.9: Translation Example

Source	[많은] [친구 들 이] [정거장 에] [마중나왔다] [.]
Gloss	[many] [friend PLURAL NOMINATIVE] [station at] [came-to-see] [.]
EBMT(SPA)	the station is a lot of friends .
EBMT(Chunk)	at the station is a lot of my friends .
Reference	many friends came down to see me at the station .

Table A.10: Translation Example

Bibliography

Steven Abney. Parsing by chunks. In *Principle-Based Parsing*, pages 257–278. Kluwer Academic Publishers, 1991.

Al-Adhaileh and Enya Kong Tang. Example-based machine translation based on the synchronous sstc, 1999.

Tantely Andriamanankasina, Kenji Araki, and Koji Tochinai. Example-based machine translation of part-of-speech tagged sentences by recursive division, 1999.

Eiji Aramaki and Sadao Kurohashi. Example-based machine translation using structural translation examples, 2004. URL `http://www.slc.atr.jp/IWSLT2004/proceedings/EC_utokyo.pdf`.

Satanjeev Banerjee and Alon Lavie. Meteor: An automatic metric for mt evaluation with improved correlation with human judgements. In *Porceedings of the Workshop on Intrinsic and Extrinsic Evaluation Measures for MT and/or Summarization at the 43rd Annual Meeting of the Association for Computational Linguistics (ACL-2005)*, June 2005.

Peter F. Brown, J. Cocke, Stephan A. Della Pietra, Vincent J. Della Pietra, F. Jelinek, Robert Mercer, and P. Roossin. A statistical approach to language translation. In *Proceedings of the 12th conference on Computational linguistics*, pages 71–76, Morristown, NJ, USA, 1988. Association for Computational Linguistics. ISBN 963 8431 56 3. doi: http://dx.doi.org/10.3115/991635.991651.

Peter F. Brown, Stephan A. Della Pietra, Vincent J. Della Pietra, and Robert Mercer. The mathematics of machine translation: Parameter estimation. *Computational Linguistics*, 19(2), 1993.

Ralf Brown. Example-based machine translation at carnegie mellon university. *The ELRA Newsletter*, 5(1), January-March 2000a. ISSN 1026-8200.

Ralf D. Brown. Example-based machine translation in the PANGLOSS system. In *Proceedings of the Sixteenth International Conference on Computational Linguistics*, pages 169–174, Copenhagen, Denmark, 1996. URL `http://www.cs.cmu.edu/~ralf/papers.html`.

Ralf D. Brown. Automated dictionary extraction for "knowledge-free" example-based translation. In *Proceedings of the Seventh International Conference on Theoretical and Methodological Issues in Machine Translation (TMI-97)*, pages 111–118, Santa Fe, New Mexico, July 1997. URL `http://www.cs.cmu.edu/~ralf/papers.html`.

Ralf D. Brown. Automated generalization of translation examples. In *Proceedings of the Eighteenth International Conference on Computational Linguistics (COLING-2000)*, pages 125–131, 2000b.

Ralf D. Brown. A modified burrows-wheeler transform for highly-scalable example-based translation. In *Machine Translation: From Real Users to Research, Proceedings of the 6th Conference of the Association for Machine Translation in the Americas (AMTA-2004)*, volume 3265 of *Lecture Notes in Artificial Intelligence*, pages 27–36. Springer Verlag, September-October 2004. URL `http://www.cs.cmu.edu/~ralf/papers.html`.

Ralf D. Brown. Context-sensitive retrieval for example-based machine translation. In *Proceedings of Workshop: Example-Based Machine Translation, The Tenth Machine Translation Summit*, pages 12–16, September 2005. URL `http://www.cs.cmu.edu/~ralf/papers.html`.

Andrew J. Carlson, Chad M. Cumby, Jeff Rosen, and Dan Roth. The SNoW learning architecture. Technical Report UIUCDCS-R-99-2101, UIUC Computer Science Department, May 1999. URL `http://l2r.cs.uiuc.edu/ danr/Papers/CCRR99.pdf`.

Pi-Chuan Chang, Michel Galley, and Christopher D. Manning. Optimizing chinese word segmentation for machine translation performance. In *StatMT '08: Proceedings of the Third Workshop on Statistical Machine Translation*, pages 224–232, Morristown, NJ, USA, 2008. Association for Computational Linguistics. ISBN 978-1-932432-09-1.

David Chiang. A hierarchical phrase-based model for statistical machine translation. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL-05)*, pages 263–270, June 2005.

David Chiang, Adam Lopez, Nitin Madnani, Christof Monz, Philip Resnik, and Michael Subotin. The hiero machine translation system: Extensions, evaluation, and analysis. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, pages 779–786, Vancouver, British Columbia, Canada, October 2005. Association for Computational Linguistics. URL `http://www.aclweb.org/anthology/H/H05/H05-1098`.

Thomas M. Cover and Joy A. Thomas. *Elements of information theory*. Wiley-Interscience, New York, NY, USA, 1991. ISBN 0-471-06259-6.

Lee R. Dice. Measures of the amount of ecologic association between species. *Journal of Ecology*, 26:297–302, 1945.

Victoria Fossum, Kevin Knight, and Steven Abney. Using syntax to improve word alignment precision for syntax-based machine translation. In *StatMT '08: Proceedings of the Third Workshop on Statistical Machine Translation*, pages 44–52, Morristown, NJ, USA, 2008. Association for Computational Linguistics. ISBN 978-1-932432-09-1.

Cyril Goutte, Kenji Yamada, and Eric Gaussier. Aligning words using matrix factorisation. In *ACL '04: Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics*, page 502, Morristown, NJ, USA, 2004. Association for Computational Linguistics. doi: http://dx.doi.org/10.3115/1218955.1219019.

T. R. G. Green. The necessity of syntax markers: Two experiments with artificial languages. *Journal of Verbal Learning and Behavior*, 18:481–496, 1979.

Yifan He and Andy Way. Improving the objective function in minimum error rate training, 2009.

Sanjika Hewavitharana, Stephan Vogel, and Alex Waibel. Augmenting a statistical translation system with a translation memory. In *Proceedings of the Tenth Workshop of the European Assocation for Machine Translation (EAMT-05)*, May 2005.

Christopher Hogan. Embedded spelling correction for ocr with an application to minority languages. In *In Workshop on Embedded MT Systems: Design, Construction, and Evaluation of Systems with an MT Component, Held in Conjunction with Association for Machine Translation in the Americas (AMTA '98)*, October 1998.

W. John Hutchins. Machine translation: A concise history, 2007.

Young-Sook Hwang, Kyounghee Paik, and Yutaka Sasaki. Bilingual knowledge extraction using chunk alignment. In *PACLIC 18*, December 2004.

Karen Sparck Jones. A statistical interpretation of term specificity and its application in retrieval. *Journal of Documentation*, 28:11–21, 1972.

Jae Dong Kim and Stephan Vogel. Iterative refinement of lexicon and phrasal alignment. In *In Machine Translation Summit XI*, pages 281–288, September 2007.

Jae Dong Kim, Ralf D. Brown, Peter J. Jansen, and Jaime G. Carbonell. Symmetric probabilistic alignment for example-based translation. In *Proceedings of the Tenth Workshop of the European Assocation for Machine Translation (EAMT-05)*, May 2005.

Dan Klein and Christopher D. Manning. Accurate unlexicalized parsing. In *IN PROCEEDINGS OF THE 41ST ANNUAL MEETING OF THE ASSOCIATION FOR COMPUTATIONAL LINGUISTICS*, pages 423–430, 2003a.

Dan Klein and Christopher D. Manning. Fast exact inference with a factored model for natural language parsing. In *In Advances in Neural Information Processing Systems 15 (NIPS)*, pages 3–10. MIT Press, 2003b.

Philipp Koehn. Pharaoh: a beam search decoder for phrase-based statistical machine translation. In *Machine Translation: From Real Users to Research, Proceedings of the 6th Conference of the Association for Machine Translation in the Americas (AMTA-2004)*, volume 3265 of *Lecture Notes in Artificial Intelligence*. Springer Verlag, September 2004a.

Philipp Koehn. Statistical significance tests for machine translation evaluation. In *EMNLP*, pages 388–395, 2004b.

Philipp Koehn and Kevin Knight. ChunkMT: Statistical machine translation with richer linguistic knowledge, 2002. URL `citeseer.ist.psu.edu/koehn02chunkmt.html`.

Philipp Koehn, 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. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL 2007)*, June 2007. demonstration session.

Philippe Langlais and Fabrizio Gotti. Ebmt by tree-phrasing. *Machine Translation*, 20(1): 1–23, 2006. ISSN 0922-6567. doi: http://dx.doi.org/10.1007/s10590-006-9017-3.

Sun Le, Jin Youbing, Du Lin, and Sun Yufang. Word alignment of english-chinese bilingual corpus based on chunks, 2000. URL `citeseer.ist.psu.edu/le02word.html`.

Zhanyi Liu, Haifeng Wang, and Hua Wu. Example-based machine translation based on tree—string correspondence and statistical generation. *Machine Translation*, 20(1):25–41, 2006. ISSN 0922-6567. doi: http://dx.doi.org/10.1007/s10590-006-9016-4.

Zhanyi Liu, Haifeng Wang, Hua Wu, and Sheng Li. Improving statistical machine translation with monolingual collocation. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 825–833, Uppsala, Sweden, July 2010. Association for Computational Linguistics. URL `http://www.aclweb.org/anthology/P10-1085`.

Yanjun Ma, Nicolas Stroppa, and Andy Way. Alignment-guided chunking. In *Proceedings of the 11th International Conference on Theoretical and Methodological Issues in Machine Translation*, pages 114–121, 2007.

Daniel Marcu and William Wong. A phrase-based, joint probability model for statistical machine translation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP-2002)*, July 2002. URL `http://www.isi.edu/~marcu/papers.html`.

Hiroshi Maruyama and Hideo Watanabe. Tree cover search algorithm for example-based translation. In *Proceedings of the Fourth International Conference on Theoretical and Methodological Issues in Machine Translation*, pages 173–184, Montreal, Canada, 1992. (TMI-92).

I. Dan Melamed. A word-to-word model of translational equivalence. In *35th Annual Meeting of the Association for Computational Linguistics (ACL'97)*, pages 490–497, 1997.

I. Dan. Melamed. Models of translational equivalence among words. *Computational Linguistics*, 26(2):221–249, 2000.

I. Dan Melamed. *Empirical Methods for Exploiting Parallel Text*. MIT Press, 2001.

Makoto Nagao. A framework of a mechanical translation between japanese and english by analogy principle. In *Proc. of the international NATO symposium on Artificial and human intelligence*, pages 173–180, New York, NY, USA, 1984. Elsevier North-Holland, Inc. ISBN 0-444-86545-4.

Franz Josef Och. Minimum error rate training in statistical machine translation, 2003.

Franz Josef Och and Hermann Hey. The alignment template approach to statistical machine translation. *Computational Linguistics*, 30(4):417–449, 2004.

Franz Josef Och and Hermann Ney. Improved statistical alignment models. In *Proceedings of the 38th Annual Meeting of the Association for Computational Linguistics (ACL-2000)*, pages 440–447, 2000.

Franz Josef Och and Hermann Ney. A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29(1):19–51, 2003. ISSN 0891-2017. doi: http://dx.doi.org/10.1162/089120103321337421.

Brigitte Orliac and Mike Dillinger. 1 collocation extraction for machine translation, 2003.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, July 2002. URL `http://acl.ldc.upenn.edu/P/P02/`.

A.B. Phillips and R.D. Brown. Cunei machine translation platform: System description. In *Proc. of the 3rd Workshop on Example-Based Machine Translation*, pages 29–36, Dublin, Ireland, Nov. 2009.

Christopher Quirk and Arul Menezes. Dependency treelet translation: the convergence of statistical and example-based machine-translation? *Machine Translation*, 20(1):43–65, 2006.

Satoshi Sato. CTM: An example-based translation aid system using the character-based best match retrieval method. In *Proceedings of the Fourteenth International Conference on Computational Linguistics (COLING-92)*, 1992.

Satoshi Sato and Makoto Nagao. Toward memory-based translation, 1990.

Helmut Schmid. Probabilistic part-of-speech tagging using decision trees. In *Proceedings of the International Conference on New Methods in Language Processing*, Manchester, UK, 1994.

Helmut Schmid. Improvements in part-of-speech tagging with an application to german. In *In Proceedings of the ACL SIGDAT-Workshop*, pages 47–50, 1995.

Kwangseob Shim and Jaehyung Yang. Mach: A supersonic korean morphological analyzer. URL `citeseer.ist.psu.edu/541656.html`.

Michel Simard, Nicola Cancedda, Bruno Cavestro, Marc Dymetman, Eric Gaussier, Cyril Goutte, Kenji Yamada, Philippe Langlais, and Arne Mauser. Translating with non-contiguous phrases. In *HLT '05: Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, pages 755–762, Morristown, NJ, USA, 2005. Association for Computational Linguistics. doi: http://dx.doi.org/10.3115/1220575.1220670.

Smadja, Frank, Kathleen R. Mckeown, and Vasileios Hatzivassiloglou. Translating collocations for bilingual lexicons: A statistical approach. *Computational Linguistics*, 22(1): 1–38, 1996.

Eiichiro Sumita and Hitoshi Iida. Experiments and prospects of example-based machine translation. In *ACL '91*, 1991.

Ben Taskar, Simon Lacoste-Julien, and Dan Klein. A discriminative matching approach to word alignment. In *HLT '05: Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, pages 73–80, Morristown, NJ, USA, 2005. Association for Computational Linguistics. doi: http://dx.doi.org/10.3115/1220575.1220585.

Tony Veale and Andy Way. Gaijin: A template-driven bootstrapping approach to example-based machine translation. In *Proceedings of the NeMNLP'97, New Methods in Natural Language Processessing*, Sofia, Bulgaria, September 1997. URL `http://-www.compapp.dcu.ie/˜tonyv/papers/gaijin.html`.

Stephan Vogel. PESA: Phrase pair extraction as sentence splitting. In *Proceedings of the Tenth Machine Translation Summit (MT Summit X)*, 2005.

Stephan Vogel, Hermann Ney, and C. Tillmann. Hmm-based word alignment in statistical translation. In *Proceedings of the 34th Annual Meeting of the Association for Computational Linguistics (ACL 1996)*, pages pp. 836–841. ACL'96, 1996.

Taro Watanabe, Eiichiro Sumita, and Hiroshi G. Okuno. Chunk-Based statistical translation. In Erhard Hinrichs and Dan Roth, editors, *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*, pages 303–310, Sapporo, Japan, 2003. URL `http://www.aclweb.org/anthology/P03-1039`.

Dekai Wu. Stochastic inversion transduction grammars and bilingual parsing of parallel corpora. *Comput. Linguist.*, 23(3):377–403, 1997. ISSN 0891-2017.

K. Yamada and K. Knight. A decoder for syntax-based statistical MT. In *ACL '02*, 2002.

Ying Zhang. Chinese word segmenter. URL `http://-projectile.is.cs.cmu.edu/research/public/tools/-segmentation/lrsegmenter/lrSegmenter.perl`.

Bing Zhao and Stephan Vogel. A generalized alignment-free phrase extraction. In *Proceedings of the ACL Workshop on Building and Using Parallel Texts*, pages 141–144, Ann Arbor, Michigan, June 2005. Association for Computational Linguistics. URL `http://www.aclweb.org/anthology/W/W05/W05-0825`.

Bing Zhao and Alex Waibel. Learning a log-linear model with bilingual phrase-pair features for statistical machine translation. In *Proceedings of the Fourth SIGHAN Workshop on Chinese Language Processing*. Association for Computational Linguistics, October 2005.

Yu Zhou, Chengqing Zong, and Bo Xu. Bilingual chunk alignment in statistical machine translation. In *2004 IEEE International Conference on Systems, Man and Cybernetics*, volume 2, pages 1401–1406, 2004.